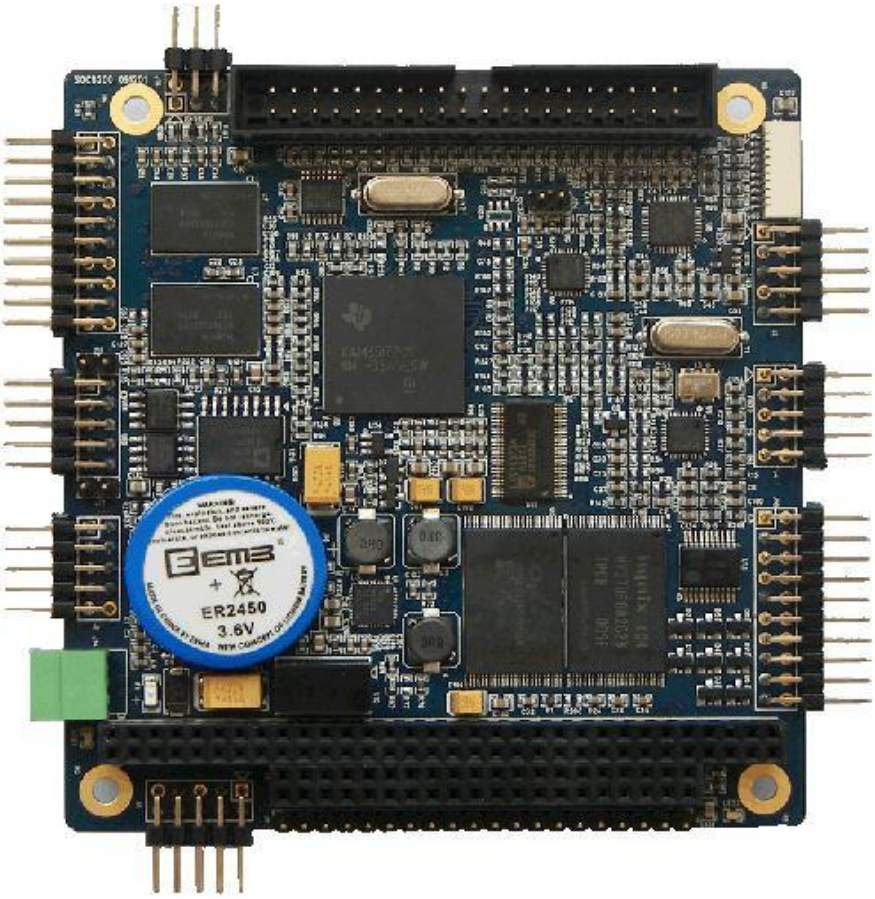
SOC8200

AM3517 Based 32-Bit Microprocessor



Quick Start Guide

Version 2.1

24th Jan 2014

A Premier Farnell Company

Copyright Statement:

- SOC8200, CAN8200, SD8200-X, CAN8200-X, AU8200, VGA8200, USB8200-X, NET8200, ECOM-4, ECOM-8, E100 Module and their related intellectual property are owned by Shenzhen Embest Technology Co., Ltd.
- AM3517 is a trademark of the TI Corporation.
- Sourcery G++ Lite for ARM GNU/Linux is trademark of Codesourcery.
- Microsoft, MS-DOS, Windows, Windows95, Windows98, Windows2000, and Windows embedded CE 6.0 are trademarks of the Microsoft Corporation.
- Shenzhen Embest Technology has the copyright of this document and reserves all rights. Any part of the document should not be modified, distributed or duplicated in any approach and form without prior written permission issued by Embest Technology Co., Ltd.

Revision History:

Version	Date	Description
1.0	10/07/2010	Original Version
2.0	28/12/2010	Modified u-boot: added NORFLASH & TV/S-video support
3.0	28/05/2011	Increased the support on WinCE 6.0
2.1	24/01/2014	Localisation

Table of Contents

1 Product Overview 1

1.1 Introduction 1

1.2 Kit Contents 1

1.3 Board Interfaces 2

1.4 System Block Diagram 3

1.5 Physical Dimensions 4

2 Hardware Features 5

2.1 SOC8200 Single Board Computer 5

2.1.1 Processor 5

2.1.2 Memory 5

2.1.3 Signals Routed to Pins 5

2.2 SOC8200 Expansion Board 6

2.2.1 Audio/Video Interfaces 6

2.2.2 Communication Interfaces 6

2.3 Operational Parameters 6

3 Hardware Details 8

3.1 Hardware Introduction 8

3.2 Hardware Interfaces 9

3.2.1 USB host & USB Interface 9

3.2.2 Network interface 10

3.2.3 Camera Interface 10

3.2.4 MMC Interface 11

3.2.5 UART1 Interface 11

3.2.6 Analog IO Interface 12

3.2.7 Digital IO Interface 12

3.2.8 TFT_LCD Interface 13

3.2.9 PC104-64 Interface 14

3.2.10 PC104-40 16
3.2.11 Multifunctional Expansion Interface 19
3.2.12 Can/485 Interface 20
3.2.13 UART (TTL) Interface 21
3.2.14 JTAG Interface 22

4 Linux System 23

4.1 Linux system Overview 23
4.1.1 Pre-installed software 23
4.1.2 BSP Features 25
4.2 Linux System Quick Operation 26
4.2.1 Windows System Environment Preparation 26
4.2.2 SOC8200 Hardware Environment Preparation 27
4.2.3 System boot methods 27
4.2.4 Display Options 28
4.3 Linux Function Tests 29
4.3.1 LED Test 29
4.3.2 Touchscreen Test 29
4.3.3 RTC Test 30
4.3.4 MMC/SD Card Test 30
4.3.5 USB OTG Test 31
4.3.6 USB HOST Test 33
4.3.7 Network Test 34
4.3.8 Camera Test 34
4.3.9 CAN Test 35
4.3.10 ADC Test 35
4.3.11 SD Card Test 36
4.3.12 Buzzer Test 36
4.3.13 Audio Test 36
4.3.14 Full-function Serial Port Test 39
4.3.15 Digital Output Test 42
4.4 Linux Image Update 42
4.4.1 Update the image on an SD card 42
4.4.2 Update the image in the NAND Flash 46

4.5 Linux System Deveplopment 48
4.5.1 Install the cross compilation environment 48
4.5.2 The installation of other tools 48
4.5.3 Adding environment variable 49
4.5.4 System complation 49
4.6 Linux System Customization 51
4.6.1 4 Alteration of kernel configuration 51
4.6.2 Compilation 53
4.6.3 Test 53
4.7 Linux Application Development 55
4.7.1 LED application development 55
5 WinCE System 57
5.1 WinCE System Overview 57
5.1.1 Pre-compiled image 57
5.1.2 Board Support Package (BSP) 59
5.2 WinCE System Quick Start 61
5.2.1 System boot 61
5.2.2 Boot from NAND Flash 61
5.2.3 Boot from SD card 61
5.3 WinCE System Development 62
5.3.1 Development environment building 62
5.4 WinCE System complation 62
5.4.1 Preparation 63
5.4.2 System Compliation 64
5.4.3 System Customization 65
5.5 WinCE image update 67
5.5.1 Updating the image on the SD card 67
5.5.2 Updating the image in NAND Flash 67
5.6 Application development 68
5.6.1 API and Application Demonstrations 70
5.6.2 The definition and demonstration of GPIO interface 70

Appendix 1: Driver Installation Of Linux USB Ethernet/RNDIS Gadget 73

Appendix 2: Making a Linux Boot Disk 76

Appendix 3: TFTP Server Setup 81

Appendix 4: WinCE related resource links 83

Appendix 5: Expansion Board 85

5.1 SOC8200-M Evaluation Suite.....85

Appendix 6: Interface Board 87

Appendix 7: ESD Precautions & Handling Procedures..... 88

Appendix 8: Technical support & Warranty 89

8.1 Technical support service.....89

8.2 Maintenance service clause.....90

8.3 Basic guidelines for protection and maintenance of LCDs 91

8.4 Value Added Services.....92

1 Product Overview

The document details development with the SOC8200, the hardware specification, features, and software development.

1.1 Introduction

The SOC8200 is an industrial evaluation kit designed and manufactured by Embest Technology Co., Ltd. The SOC8200 is based on the Texas Instruments AM3517 processor. This processor is integrated with a 600Mhz ARM Cortex-A8 Core which is used as a dedicated industrial signal processor. The SOC8200 provides: 10/100Mbps Ethernet interface, S-VIDEO interface, Audio input and output interfaces, USB device, USB HOST, SD card interface, series port, CF card, SPI interface, I2C interface, JTAG interface, CAMERA interface, LCD interface, touchscreen interface and keyboard as well as an HDMI (DVI-D) interface. Its high performance and low power consumption enable the device to support the following applications:

- Industrial control;
- Field communication;
- Medical equipment;
- Instrumentation;
- Security systems.

1.2 Kit Contents

- ✓ SOC8200 Single Board Computer
- ✓ SOC8200 Expansion Board
- ✓ 2GB SD card
- ✓ Serial cable
- ✓ Ethernet cable
- ✓ 12V power adaptor
- ✓ 2x5 pin, 12cm cable
- ✓ 2x20 pin, 12cm cable
- ✓ CAN8200 module
- ✓ CD including software and documentation
- ✓ **Optional** LCD touchscreen 4.3” (480x272) **or** 7” (800x480)

1.3 Board Interfaces

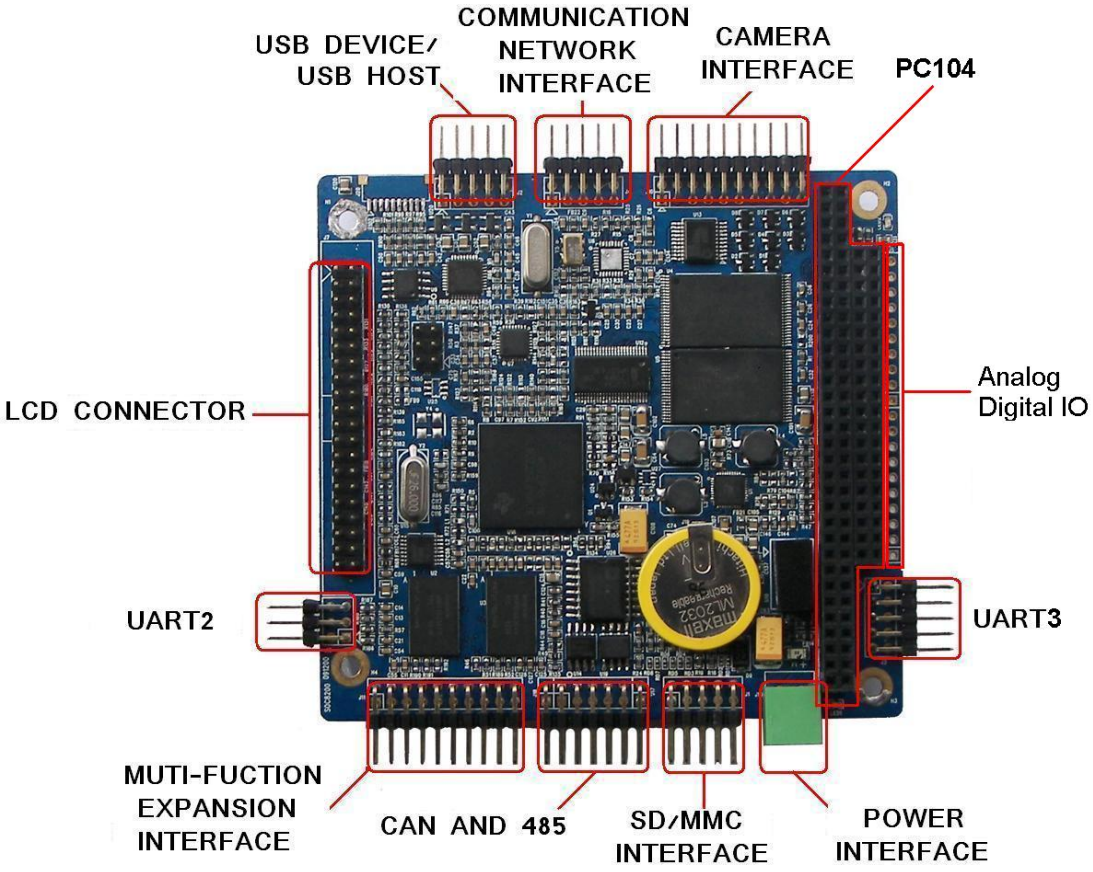


Figure 1: SOC8200 Board Interfaces

1.4 System Block Diagram

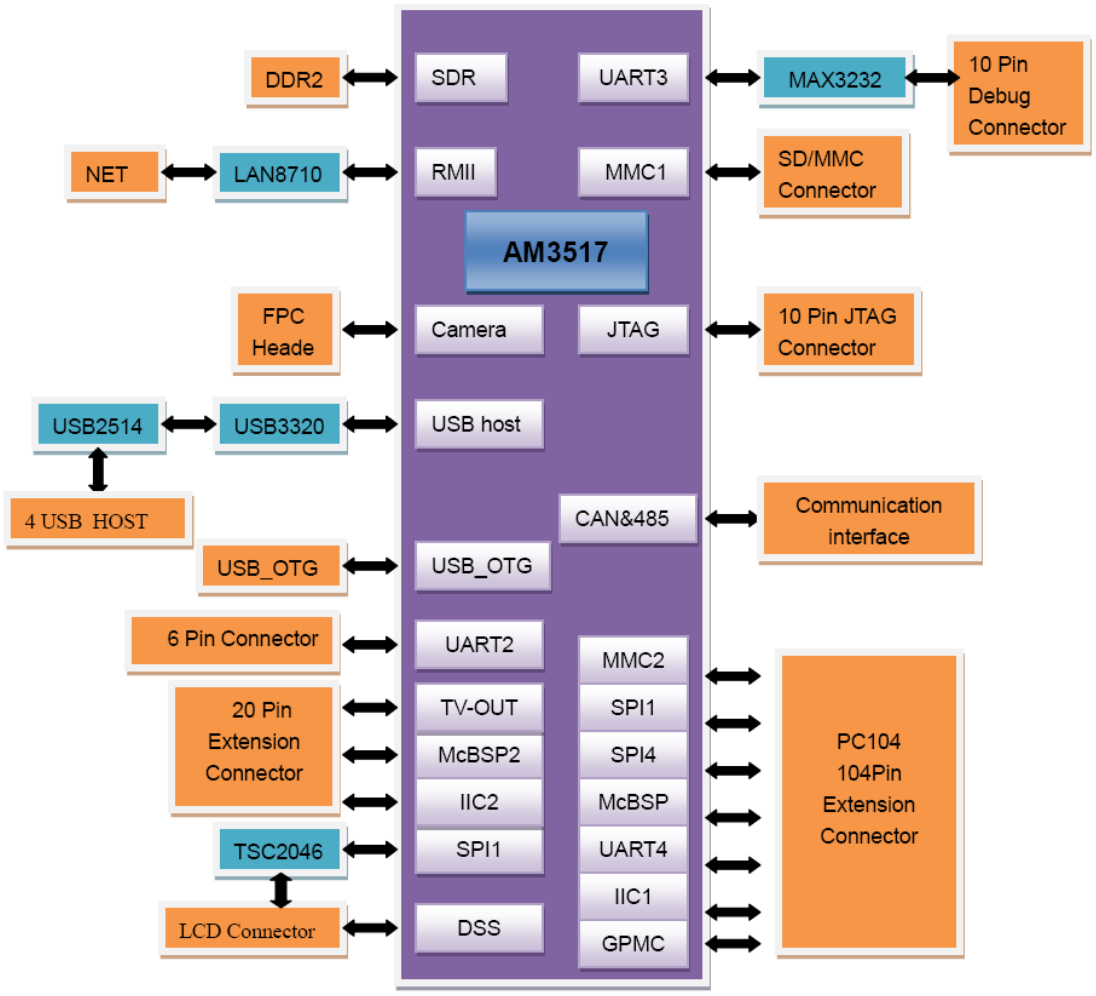


Figure 2: SOC8200 System Block Diagram

1.5 Physical Dimensions

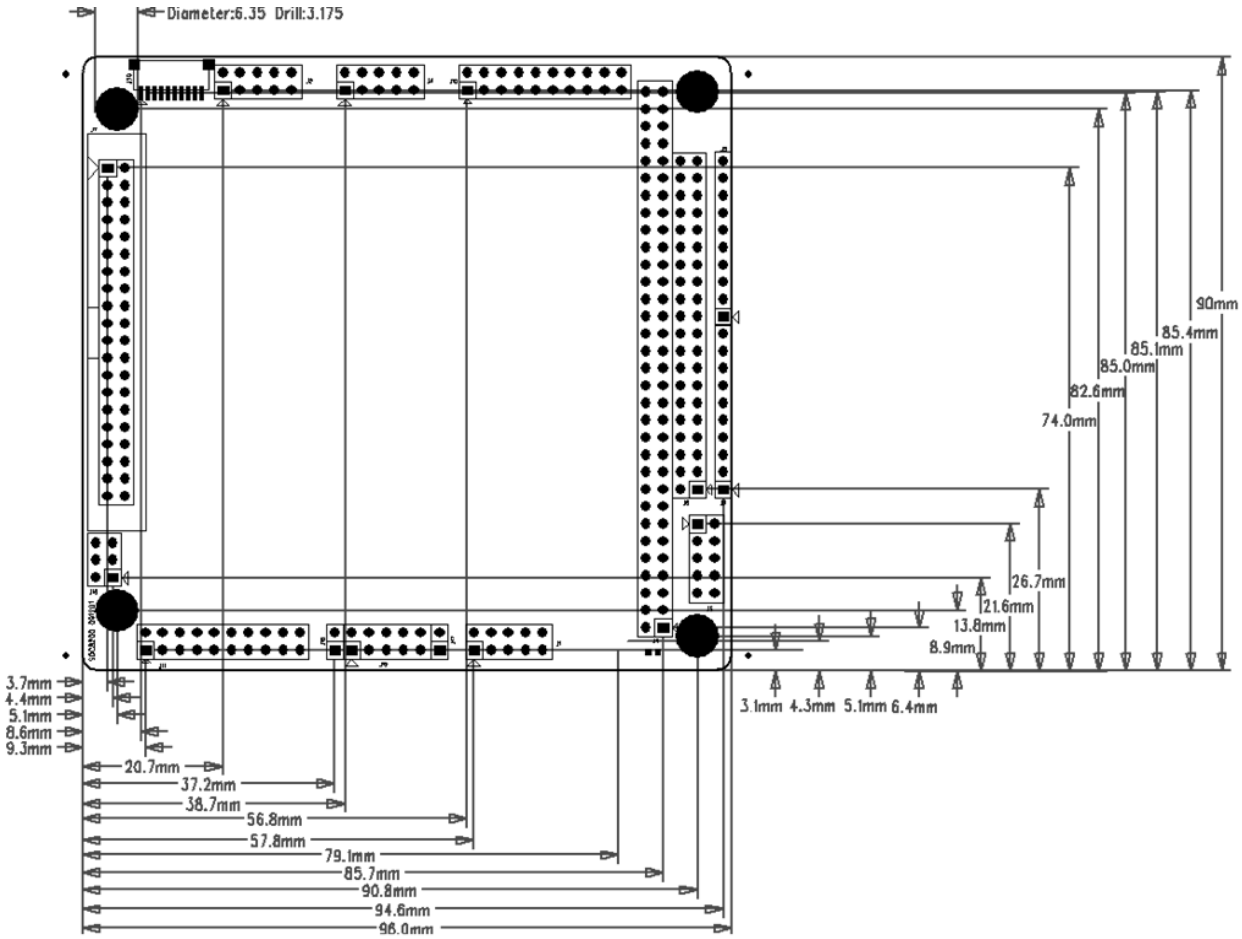


Figure 3: SOC8200 Physical Dimensions

2 Hardware Features

2.1 SOC8200 Single Board Computer

2.1.1 Processor

- AM3517 industrial applications processor
- 600MHz ARM Cortex-A8 Core
- NEON SIMD Coprocessor
- POWERVR SGX Graphics Accelerator
- 16KB I-Cache,
- 16KB D-Cache,
- 256KB L2-Cache,
- 112KB ROM,
- 64KB Shared SRAM

2.1.2 Memory

- 256MB DDR2 SDRAM, 32bit
- 256MB NAND Flash, 8bit
- 4MB NOR Flash, 16bit

2.1.3 Signals Routed to Pins

- 5-wire Debug serial port (RS232)
- 5-wire serial port (TTL)
- Two USB 2.0 Host High-speed ports, 480Mbps
- USB 2.0 Device High-speed port, 480Mbps
- Audio (IIS)
- 16-bit LCD output
- 10-bit Camera video input
- 1-channel S-Video output
- 1-channel AV output
- RS485 serial port

- 1-channel CAN bus
- 10/100Mbps Ethernet port
- SD/MMC card
- Multi-functional expansion interface (McBSP, IIC, McSPI, TV-OUT)
- PC104 expansion interface (GPMC Bus, MMC, USB, McSPI, UART1, Clock, HDQ)
- JTAG interface

2.2 SOC8200 Expansion Board

2.2.1 Audio/Video Interfaces

- Audio input port
- Stereo audio output port
- 15-pin standard VGA output interface
- Buzzer

2.2.2 Communication Interfaces

- 5-wire RS232 serial port (DB9)
- 9-wire RS232 serial port (DB9)
- 9-wire TTL serial port (2x5 pin 2.5 mm pitch connector)
- Two High-speed USB 2.0 Host ports, 480Mbps
- High-speed USB 2.0 Device port, 480Mbps
- 10/100Mbps Ethernet port (RJ45)
- 10-bit Camera interface
- Reset button
- SD/MMC card slot
- CF card slot

2.3 Operational Parameters

- Dimensions: 96x90 mm (Core board)
- 170x190 mm (Expansion board)

- Input Voltage: +5V (Core board)
- +12V (Expansion board)
- Layers: 8
- Working Temperature: -40 ~ 85°C

3 Hardware Details

3.1 Hardware Introduction

The SOC8200 board has onboard:

- 256MB DDR2 SDRAM,
- 256MB Nand Flash, and
- 4MB Nor Flash

It extends various functions through pins including:

- Serial port,
- Ethernet,
- CAN,
- RS485,
- SD/MMC card,
- CF card,
- Audio In/Out,
- Camera,
- LCD,
- USB Host,
- USB Device,
- Expansion connector and
- JTAG.

Embest has designed an expansion board and several function interface boards for the SOC8200. The expansion board makes the ideal companion to the SOC8200 for evaluating the functionality of the Texas Instruments' Sitara AM3517 microprocessor. In addition, functions can be added through several choices of function interface boards according to your requirements which can effectively shorten the period of research and development of products and considerably speed up the time to market.

For a detailed schematic of the SOC8200 please refer to:

http://www.armkits.com/download/soc8200sch.pdf

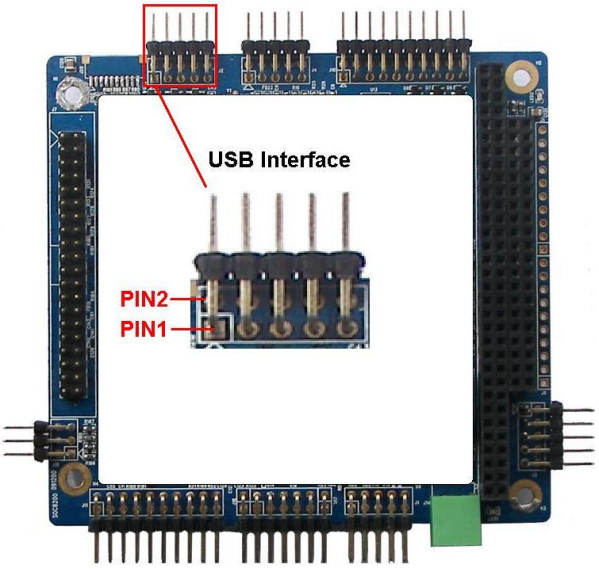
Note:

The schematic is only for reference, we do not offer any technical support related to its use or development.

3.2 Hardware Interfaces

3.2.1 USB host & USB Interface

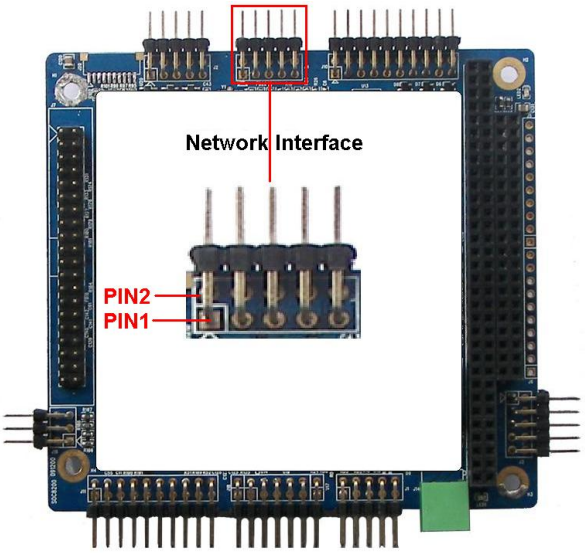
PIN	Description
1	OTG_DM
2	OTG_DP
3	OTG_BUS
4	OTG_ID
5	U1_DM
6	U1_DP
7	GND
8	POWER_USB
9	U2_DM
10	U2_DP



3.2.2 Network interface

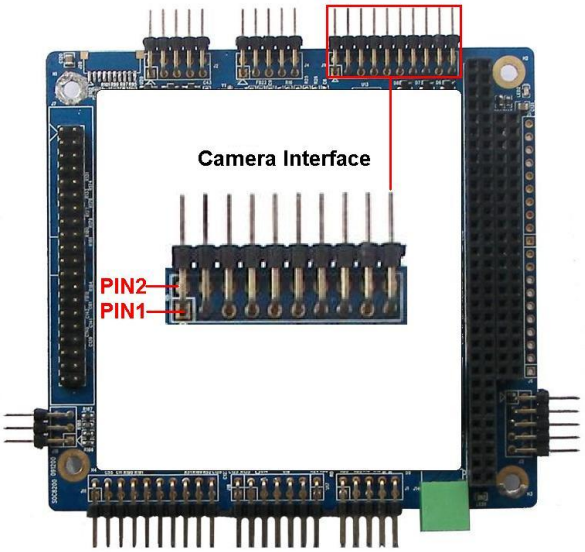
SOC8200 is a 10M/100M adaptive network interface

PIN	Description
1	GND
2	VCC_IO
3	TXN
4	TXP
5	GND
6	RXN
7	RXP
8	LED2/NINTSEL
9	LED1/REGOFF
10	GND



3.2.3 Camera Interface

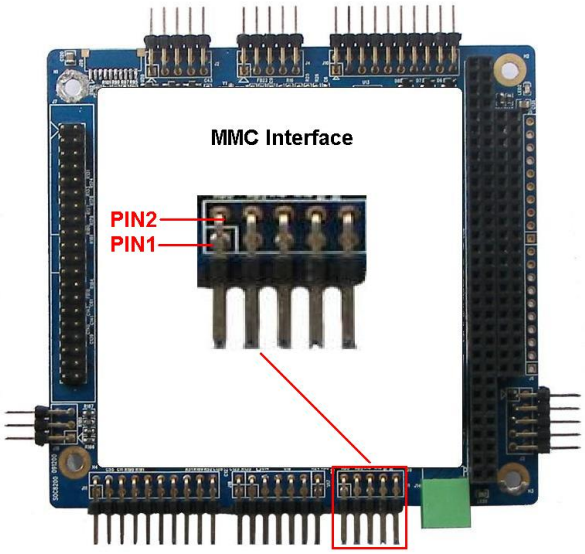
PIN	Description
1	GND
2	CAM_D0
3	CAM_D1
4	CAM_D2
5	CAM_D3
6	CAM_D4
7	CAM_D5
8	CAM_D6
9	CAM_D7
10	CAM_D8
11	CAM_D9



PIN	Description
12	GND
13	CAM_PCLK
14	GND
15	CAM_HS
16	CAM_VS
17	VCC_IO
18	IIC3_SDA
19	IIC3_SCL
20	GND

3.2.4 MMC Interface

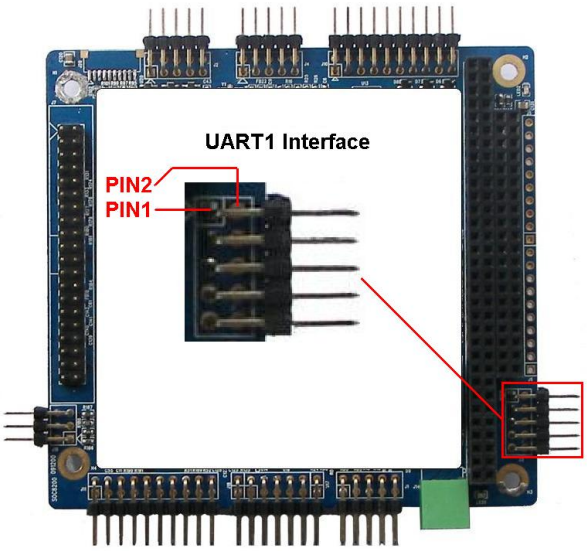
PIN	Description
1	VCC_IO
2	MMC1_CLK
3	MMC1_CMD
4	MMC1_D0
5	MMC1_D1
6	MMC1_D2
7	MMC1_D3
8	MMC1_CD
9	MMC1_WP
10	GND



3.2.5 UART1 Interface

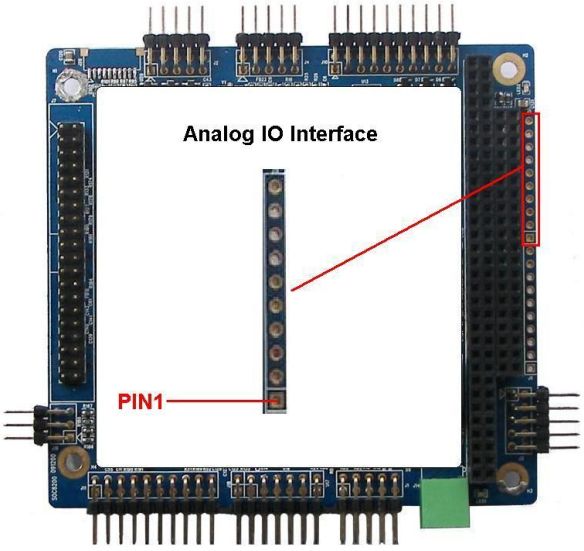
PIN	Description
1	N/A

2	N/A
3	R1IN
4	T2OUT
5	T1OUT
6	R2IN
7	N/A
8	N/A
9	GND
10	N/A



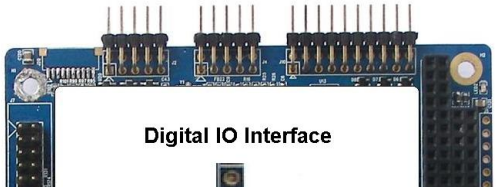
3.2.6 Analog IO Interface

PIN	Description
1	CH7
2	CH6
3	CH5
4	CH4
5	CH3
6	CH2
7	CH1
8	CH0
9	GND
10	VCC_IO



3.2.7 Digital IO Interface

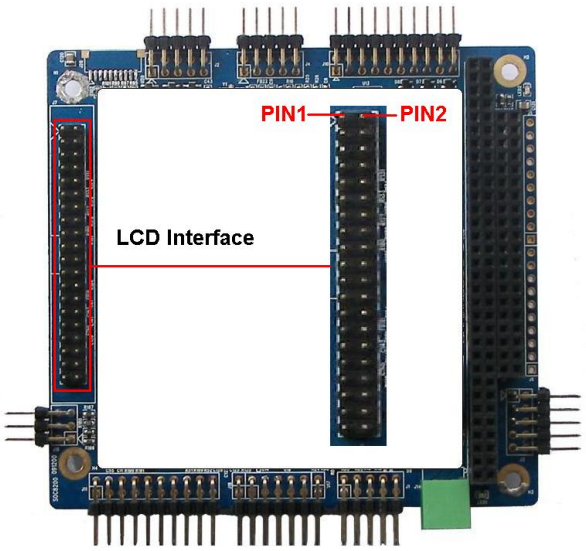
PIN	Description
1	CH7
2	CH6



3	CH5
4	CH4
5	CH3
6	CH2
7	CH1
8	CH0
9	GND
10	VCC_IO

3.2.8 TFT_LCD Interface

PIN	Description
1	GND
2	DSS_CLK
3	DSS_HS
4	DSS_VS
5	GND
6	N/A
7	DSS_D11
8	DSS_D12
9	DSS_D13
10	DSS_D14
11	DSS_D15
12	GND
13	DSS_D5
14	DSS_D6
15	DSS_D7
16	DSS_D8
17	DSS_D9

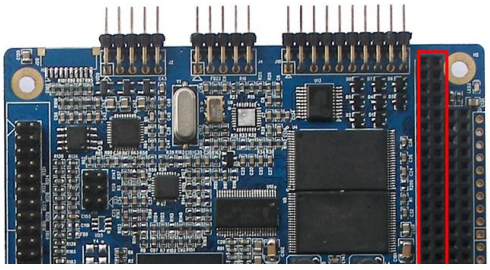


PIN	Description
37	VCC_5V
38	LCD_ADJ
39	GND
40	N/A

PIN	Description
18	DSS_D10
19	GND
20	N/A
21	DSS_D0
22	DSS_D1
23	DSS_D2
24	DSS_D3
25	DSS_D4
26	GND
27	DSS_DEN
28	VCC_IO
29	VCC_IO
30	N/A
31	N/A
32	Y+
33	X-
34	Y-
35	X+
36	LCD_PEN

3.2.9 PC104-64 Interface

PIN	Description
1	GND
2	N/A
3	VCC_5V
4	VCC_5V



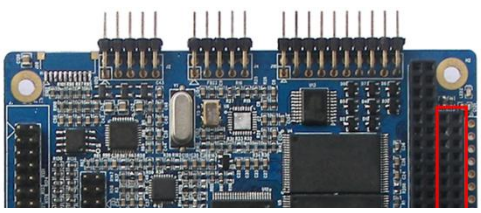
PIN	Description
5	VCC_IO
6	VCC_IO
7	GND
8	SYS_RST
9	POWER_RST
10	SYS_CLKOUT2
11	SYS_CLKOUT1
12	HDQ_SIO
13	SYS_32K
14	GND
15	IRQ
16	GPIO58
17	GPIO57
18	GPIO56
19	GPT11
20	GPT10
21	GPT9
22	IIC1_SDA
23	IIC1_SCL
24	GND
25	UART4_RX
26	RS485_TXEN
27	RS485_RXEN
28	UART4_TX
29	GND
30	MCBSP4_DX

PIN	Description
50	MMC2_D6
51	MMC2_D5
52	MMC2_D4
53	MMC2_D3
54	MMC2_D2
55	MMC2_D1
56	MMC2_D0
57	MMC_CMD
58	MMC2_CLK

PIN	Description
31	MCBSP4_DR
32	MCBSP4_CLKX
33	MCBSP4_FSX
34	MCBSP_CLKS
35	GND
36	GPIO157
37	GPIO162
38	SPI4_CS0
39	SPI4_SOMI
40	SPI4_SIMO
41	SPI4_CLK
42	GND
43	SPI1_CS3
44	SPI1_CS2
45	SPI1_SOMI
46	SPI1_SIMO
47	SPI1_CLK
48	GND
49	MMC2_D7

3.2.10 PC104-40

PIN	Description
1	GND
2	GPMC_NCS4
3	GPMC_NCS3

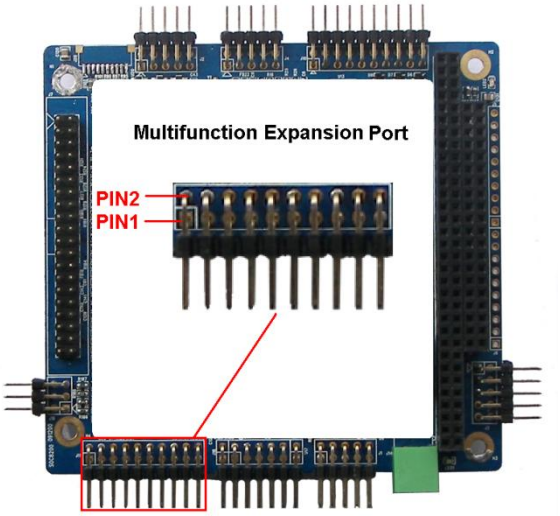


PIN	Description
4	GPMC_NCS2
5	GPMC_A10
6	GPMC_A9
7	GPMC_A8
8	GPMC_A7
9	GPMC_A6
10	GPMC_A5
11	GPMC_A4
12	GPMC_A3
13	GPMC_A2
14	GPMC_A1
15	GPMC_NBE1
16	GPMC_WAIT3
17	SYS_RST
18	GPMC_CLE
19	GND
20	GPMC_ALE
21	GPMC_CLK
22	GPMC_WE
23	GPMC_OE
24	GPMC_D15
25	GPMC_D14
26	GPMC_D13
27	GPMC_D12
28	GPMC_D11
29	GPMC_D10

PIN	Description
30	GPMC_D9
31	GPMC_D8
32	GPMC_D7
33	GPMC_D6
34	GPMC_D5
35	GPMC_D4
36	GPMC_D3
37	GPMC_D2
38	GPMC_D1
39	GPMC_D0
40	GND

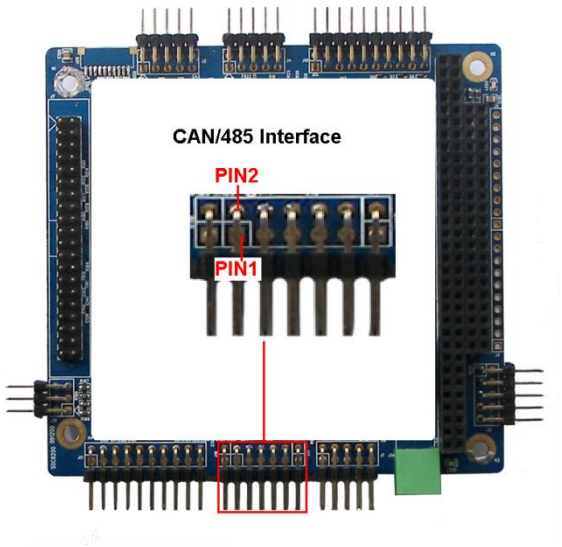
3.2.11 Multifunctional Expansion Interface

PIN	Description
1	GND
2	TV_OUT1
3	GND
4	TV_OUT2
5	GND
6	VCC_IO
7	GND
8	MCBSP2_CLKX
9	MCBSP2_FSX
10	MCBSP2_DR
11	MCBSP2_DX
12	IIC2_SDA
13	IIC2_SCL
14	GND
15	SPI2_CLK
16	SPI2_SIMO
17	SPI2_SOMI
18	SPI2_CS0
19	SPI2_CS1
20	GND



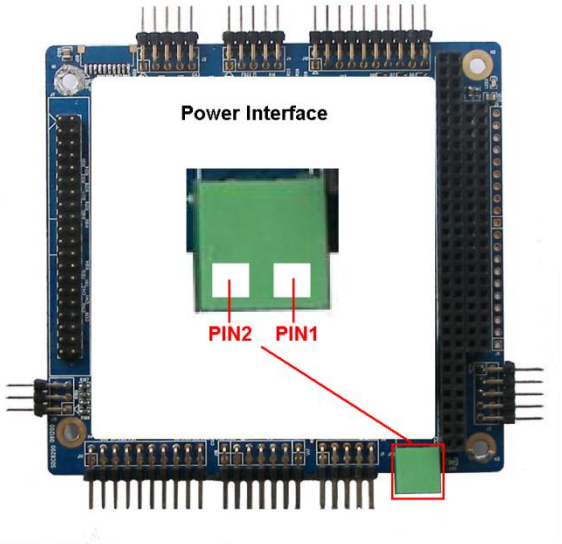
3.2.12 Can/485 Interface

PIN	Description
1	CANH
2	CANH
3	CANL
4	CANL
5	CHGND
6	CHGND
7	RS485A
8	RS485B
9	RS485Z
10	RS485Y
11	MCBSP2_DX
12	IIC2_SDA
13	IIC2_SCL
14	GND
15	SPI2_CLK
16	SPI2_SIMO
17	SPI2_SOMI
18	SPI2_CS0
19	SPI2_CS1
20	GND



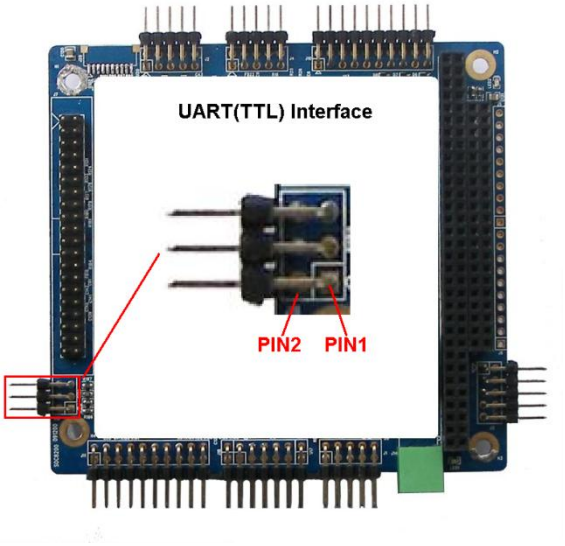
Power Interface

PIN	Description
1	VCC_5V
2	GND



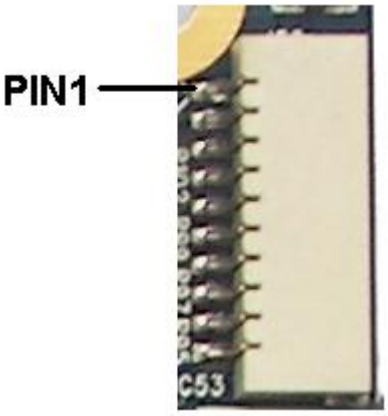
3.2.13 UART (TTL) Interface

PIN	Description
1	VCC_IO
2	GND
3	UART2_CTS
4	UART2_RTS
5	UART2_TX
6	UART2_RX



3.2.14 JTAG Interface

PIN	Description
1	VCC
2	TMS
3	TDI
4	NTRST
5	TD0
6	RTCK
7	TCK
8	EMU0
9	EMU1
10	GND



4 Linux System

4.1 Linux system Overview

This chapter provides an overview of the software system of the SOC8200, including the introduction of pre-installed software, specifications of the SOC8200 BSP (Board Support Package) and various specifications contained on the SOC8200 CD.

The SOC8200 software system includes: pre-compiled images, application system source code, cross compilation tools, auxiliary tools for development. Images, applications, source code and auxiliary tools all of which can be found in the CD.

The SD card provided with the SOC8200 has the following software:

- x-loader----------(x-load.bin.ift_for_NAND)
- u-boot----------(flash-uboot.bin)
- kernel----------(uImage)
- rootfs----------(ubi.img)

In addition, the CD provides the following programs and software:

- The image files for burning
- Cross compilation tools
- Source code for each part of system
- User testing programs and development demonstrations
- Some tools that may be used by users when operating the SOC8200

4.1.1 Pre-installed software

A software image has been preinstalled in the NAND Flash. The completed system consists of four parts: i.e. x-loader, u-boot, kernel and rootfs. Figure 4 shows the structure of the system:

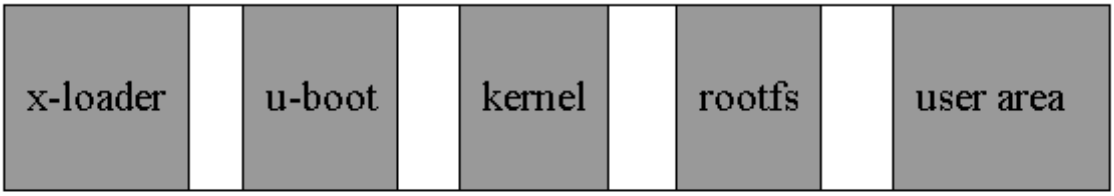


Figure 4: Software System Structure

The features and functions of each part of the system are:

1. **x-loader** is a first level bootstrap program. After the system start-up, the ROM inside the CPU will copy the x-loader to internal RAM and perform work. Its main function is to initialize the CPU, copy u-boot into the memory and give the control power to u-boot;
2. **u-boot** is a second level bootstrap program. It is used for interacting with users and updating images and leading the kernel;
3. The latest 2.6.32.x **kernel** is employed and can be customized based on the SOC8200;
4. **Rootfs** employs an open-source system. It is small in capacity and powerful.

4.1.2 BSP Features

The SOC8200 BSP is used for customizing and generating a Linux operating system applicable to the SOC8200 hardware platform. Users can conduct a secondary development on the basis of this BSP. The BSP in the CD included with the SOC8200 contains the following:

Item		Description
Bootloader	x-loader	NAND / ONENAND
		MMC / SD
		FAT
	u-boot	NAND / ONENAND
		MMC / SD
		FAT
		NET
Kernel	Linux-2.6.32	ROM/CRAM/EXT2/EXT3/FAT/NFS/ JFFS2/UBIFS
Device Driver	serial	Serial port driver
	full function serial x2	Full function serial (TTY level+RS232 level)
	RTC	Hardware clock driver
	net	10/100M Ethernet driver
	flash	NAND flash driver
	lcd	TFT LCD driver
	touch screen	Touch screen driver
	vga	Supports VGA output
	mmc/sd	MMC/SD controller driver
	cf	Support CF card device
	usb otg	USB otg 2.0 driver
	usb ehci	USB ehci driver
	video	Supports video output

Item		Description
	audio	Sound driver (supports audio Play)
	buzzer	Buzzer driver
	can	CAN driver
	led	LED driver
	rs485	RS485 driver
	Watch dog	Watch dog driver
	Analog Input	8 channel analog input
	Digital output	Control independently

4.2 Linux System Quick Operation

4.2.1 Windows System Environment Preparation

During the course of system operation, when HyperTerminal is required, the configuration is as follows:

- Baud rate: 115200
- Data bit: 8
- Parity check: no
- Stop bit: 1

4.2.2 SOC8200 Hardware Environment Preparation

Before turning on the power and booting the Linux system, you should ensure the following:

1. Confirm that you have connected the LCD/VGA if required;
2. Insert the SD card;
3. Connect the serial port to both the SOC8200 and a PC;
4. Start up a HyperTerminal window on the PC;
5. Connect the 12V power supply;
6. When the HyperTerminal window on the PC shows SOC8200's information, it confirms that the board boot up was successful.

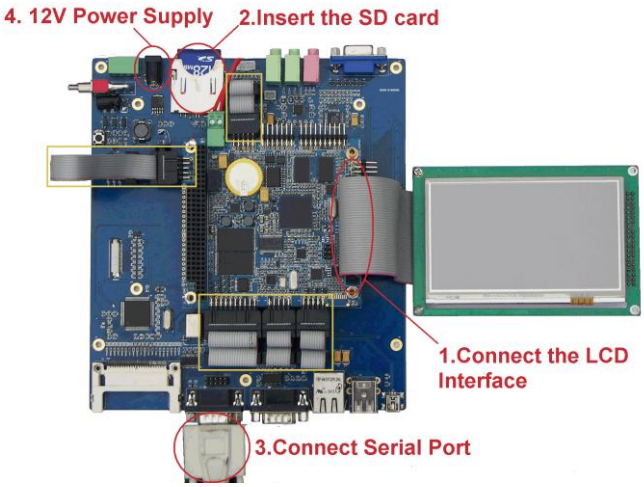


Figure 5: SOC8200 Interface Locations

4.2.3 System boot methods

NAND Flash Boot

You can use a jumper cap to choose the boot type, if you connect a jumper cap across jumper J24, the board will boot from the SD card, otherwise the board will boot from NAND Flash.

The NAND Flash has been preinstalled with the required drivers for a 4.3" screen, the user need only to connect the serial port and correctly configure a HyperTerminal connection.

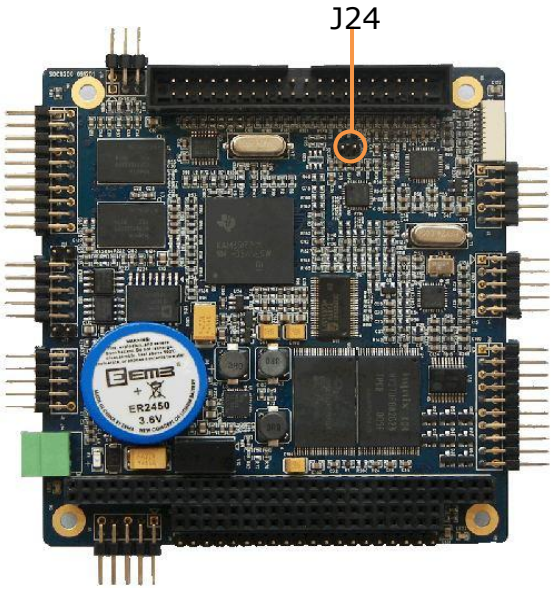


Figure 6: Jumper Location for SD Card Boot

The Linux system can be booted from the

NAND Flash without connecting the jumper cap. If you do not wish to use VGA, please refer to: Update the image in the NAND Flash on p46.

SD card Boot

If you need to boot from the SD card, please refer to the following section

4.2.4 Display Options

To enable display on LCDs (4.3”, 7”, 10.4”) or VGA, use the instructions below:

1. Replace the SD card uImage file

Copy the required image to the SD card and delete the previous uImage. Rename the “uImage_xx” to “uImage” on the SD card.

Warning :

- The XX”LCD is the size of LCD you are using, is the options included are: 4.3”, 7”, 10.4”, and VGA).

2. Enter U-boot Shell

```
40X

Texas Instruments X-Loader 1.45 (Mar 19 2010 - 16:09:58)
Starting X-loader on MMC
Reading boot sector

213544 Bytes Read from MMC
Starting OS Bootloader from MMC...
Starting OS Bootloader...



U-Boot 2009.11-svn ( 3 19 2010 - 16:14:31)

OMAP34xx/35xx-GP ES1.0, CPU-OPP2 L3-165MHz
AM3517EVM Board + LPDDR/NAND
I2C:   ready
DRAM:  256 MB
NAND:  256 MiB
In:    serial
Out:   serial
Err:   serial
Die ID #796400000000000001543b2106011005
```

```
Net:   davinci_emac_initialize
Ethernet PHY: GENERIC @ 0x00
DaVinci EMAC
Hit any key to stop autoboot:  3
```

Once at this point, the system will count down for 3 seconds. Then press any key to enter the u-boot shell.

3. U-boot parameter settings

Input the following (input characters in bold) into the u-boot shell.

```
    OMAP3517EVM # setenv bootargs console=ttyS2,115200n8 root=/dev/ram0
rw rootfstype=ext2 initrd=0x81600000,40M

    OMAP3517EVM # setenv bootcmd 'mmc init\;fatload mmc 0 80300000
uImage\;fatload mmc 0 81600000 ramdisk.gz\;bootm 0x80300000'

    OMAP3517EVM # saveenv

    OMAP3517EVM # boot
```

Reboot the system after input is complete.

4.3 Linux Function Tests

4.3.1 LED Test

Led1 on the SOC8200 has been used as a power indicator light, led2 can be used as required.

The following code can be used on Linux systems to test led2.

1. Light LED2.

```
[root@OMAP3EVM /]# echo -n 1 >/sys/class/leds/led/brightness
```

2. Extinguish LED2.

```
[root@OMAP3EVM /]# echo -n 0 >/sys/class/leds/led/brightness
```

4.3.2 Touchscreen Test

After entering the Linux system, execute the following commands to test the touchscreen:

1. Input the following command to begin the touchscreen calibration procedure:

```
[root@OMAP3EVM /]# ts_calibrate
```

Follow the prompts on the screen, click the "+" icon five times to complete the calibration.

2. After the calibration is complete, enter the following command for the touch-screen test

```
[root@OMAP3EVM /]# ts_test
```

Follow the prompts on the screen to complete testing.

4.3.3 RTC Test

The SOC8200 has a hardware clock, it is used to save and restore the system time, refer to the following test methods:

1. Set the system time as 5:55 p.m. on the 24th March 2010:

```
[root@OMAP3EVM /]# date 032417552010
Wed Mar 24 17:55:00 UTC 2010
```

2. Write the system time into RTC:

```
[root@OMAP3EVM /]# hwclock -w
```

3. Read RTC:

```
[root@OMAP3EVM /]# hwclock
Wed Mar 24 17:55:06 2010  0.000000 seconds
```

The hardware clock RTC will be set to 24th March 2010 and the system time is saved in the hardware clock.

4. Reboot the system and input the following commands to restore the system time.

```
[root@OMAP3EVM /]# hwclock -s
[root@OMAP3EVM /]# date
Wed Mar 24 17:55:37 UTC 2010
```

4.3.4 MMC/SD Card Test

1. Insert the MMC/SD card and the system will display the detection information:

```
[root@OMAP3EVM /]# mmc0: new MMC card at address 0001
```

```
mmcblk1: mmc0:0001 000000 122 MiB
mmcblk1: p1
```

2. Mount the MMC/SD card to directory /mnt:

```
[root@OMAP3EVM /]# mount -t vfat /dev/mmcblk1p1 /mnt/
[root@OMAP3EVM /]# ls /mnt MLO u-boot.bin ubi.img
flash-uboot.bin uImage x-load.bin.ift_for_NAND
```

3. Umount the SD card:

```
[root@OMAP3EVM /]# umount /mnt
```

4.3.5 USB OTG Test

Use the SOC8200 as a DEVICE and the USB OTG as a slave：

1. The user can connect the development board and the pc via a USB mini B to USB A cable after the system runs.

Note:

📖 Install the Linux USB Ethernet/RNDIS Gadget driver according to 5Appendix 1:Driver Installation Of Linux USB Ethernet/RNDIS Gadget.

2. After successful connection, the PC will show a virtual network card as displayed in below figure：

Figure 7: Virtual Network

3. Set the IP address of the virtual network card, for example:

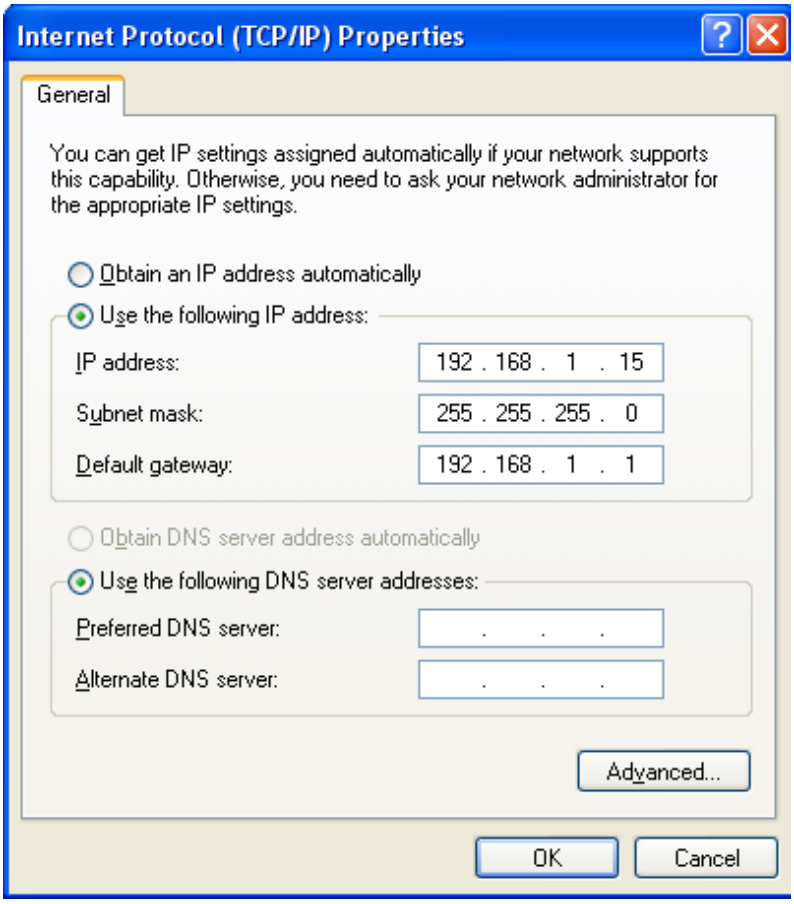

Internet Protocol (TCP/IP) Properties
General
You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.
Obtain an IP address automatically
Use the following IP address:
IP address:
192 . 168 . 1 . 15
Subnet mask:
255 . 255 . 255 . 0
Default gateway:
192 . 168 . 1 . 1
Obtain DNS server address automatically
Use the following DNS server addresses:
Preferred DNS server:
Alternate DNS server:
Advanced...
OK
Cancel

Figure 8: IP Configuration

4. Ensure that the IP network segment of the SOC8200 board is the same as the virtual network card:

```
[root@OMAP3EVM /]# ifconfig usb0 192.168.1.105
[root@OMAP3EVM /]# ifconfig
usb0      Link encap:Ethernet  HWaddr CA:A3:26:97:50:A8
          inet addr:192.168.1.105  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:145 errors:0 dropped:0 overruns:0 frame:0
          TX packets:6 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:14777 (14.4 KiB)  TX bytes:796 (796.0 B)
```

5. Use the following command on HyperTerminal to test if the SOC8200 board has connected successfully.

```
 [root@OMAP3EVM /]# ping 192.168.1.15
PING 192.168.1.15 (192.168.1.15): 56 data bytes
64 bytes from 192.168.1.15: seq=0 ttl=128 time=6.592 ms
```

```
64 bytes from 192.168.1.15: seq=1 ttl=128 time=0.549 ms
64 bytes from 192.168.1.15: seq=2 ttl=128 time=0.488 ms
64 bytes from 192.168.1.15: seq=3 ttl=128 time=0.458 ms
```

Note:

The IP address of the OTG should not be the same as the virtual network adaptor, you may need to change it.

4.3.6 USB HOST Test

Connect the SOC8200 to a USB device:

1. Insert USB-disk into the USB-HOST interface, when the SOC8200 board runs the system shows the detection information automatically.

```
[root@OMAP3EVM /]# usb 1-1.1: new full speed USB device using ehci-omap and address
4
usb 1-1.1: not running at top speed; connect to a high speed hub
usb 1-1.1: New USB device found, idVendor=1976, idProduct=1307
usb 1-1.1: New USB device strings: Mfr=1, Product=2, SerialNumber=3
usb 1-1.1: Product: USB Reader
usb 1-1.1: Manufacturer: ChipsBnk
usb 1-1.1: SerialNumber: 110074973765
scsi1 : usb-storage 1-1.1:1.0
scsi 1:0:0:0: Direct-Access     ChipsBnk SD/MMCReader    4081 PQ: 0 ANSI: 2
sd 1:0:0:0: [sdb] 1990656 512-byte logical blocks: (1.01 GB/972 MiB)
sd 1:0:0:0: [sdb] Write Protect is off
sd 1:0:0:0: [sdb] Assuming drive cache: write through
sd 1:0:0:0: [sdb] Assuming drive cache: write through
sdb: sdb1
sd 1:0:0:0: [sdb] Assuming drive cache: write through
sd 1:0:0:0: [sdb] Attached SCSI removable disk
```

2. Mount the USB-disk to the /mnt directory:

```
 [root@OMAP3EVM /]# mount -t vfat /dev/sdb1/mnt/
```

3. Check the USB-disk information.

```
[root@OMAP3EVM /]# ls /mnt/
MLO                  ramdisk.gz            uImage-vga
emtest_auto           saMmapLoopback         ubi.img
fancuo.mp3            u-boot.bin            x-load.bin.ift_for_NAND
```

```
flash-uboot.bin         uImage
```

4. Umount the USB-host.

```
[root@OMAP3EVM /]# umount /mn
```

4.3.7 Network Test

The board has a 10/100M self-adapting network card (DM9000); users can connect the board to the LAN and enter the following commands to test:

```
[root@OMAP3EVM /]# ifconfig eth0 192.192.192.201
eth0: attached PHY driver [Generic PHY] (mii_bus:phy_addr=ffffffff:00, id=7c0f1)
[root@OMAP3EVM /]# PHY: ffffffff:00 - Link is Up - 100/Full

[root@OMAP3EVM /]# ping 192.192.192.90
PING 192.192.192.90 (192.192.192.90): 56 data bytes
64 bytes from 192.192.192.90: seq=0 ttl=128 time=5.005 ms
64 bytes from 192.192.192.90: seq=1 ttl=128 time=0.396 ms
64 bytes from 192.192.192.90: seq=2 ttl=128 time=0.305 ms
64 bytes from 192.192.192.90: seq=3 ttl=128 time=0.305 ms
64 bytes from 192.192.192.90: seq=4 ttl=128 time=0.305 ms
```

Note:

- The IP address shown above is an example, it can be changed if required
- Press ctrl+c to quit.

4.3.8 Camera Test

Connect the camera module and the CCD camera well, execute the following commands to test after entering the system.

```
[root@OMAP3EVM /mnt]# saMmapLoopback
Capture: Opened Channel
Capture: Current Input: Composite
Capture: Input changed to: Composite
Capture: Current standard: NTSCvpfe-capture vpfe-capture: width = 720, height
= 480, bpp = 2

vpfe-capture vpfe-capture: adjusted width = 720, height = 480, bpp = 2,
bytesperline = 1440, sizeimage = 691200

Capture: Number of requested buffers = 2
Capture: Init done successfully
```

```
Display: Opened Channel
Display: Capable of streaming
Display: Number of requested buffers = 3
Display: Init done succetvp514x 3-005d: tvp5146 (Version - 0x03) found at 0xba
(OMAP I2C adapter)
ssfully

Display: Stream on...
Capture: Stream on...
```

The LCD shows the image collected by the camera. (Press ctrl+c to quit the test)

4.3.9 CAN Test

If the user wants to connect a CAN device, please use the CAN8200.

The steps for CAN connection are below:

1. Set the CAN baud rate as 125 k/bits, and enable the CAN devices.

```
[root@OMAP3EVM bin]# /usr/bin/ip link set can0 type can bitrate 125000
triple-sampling on
[root@OMAP3EVM bin]# /usr/bin/ip link set can0 up
ti_hecc ti_hecc.1: setting CANBTC=0xc00a8
```

2. Input the following commands to send the data: "**1122334455667788**".

```
[root@OMAP3EVM bin]# /usr/bin/cansend can0 111#1122334455667788
```

3. Input the following commands to receive the data:

```
[root@OMAP3EVM bin]# /usr/bin/candump can0
```

Receive the data that you have sent:

```
  can0       80  [8] 01 02 03 04 05 06 07 08
  can0       80  [8] 01 02 03 04 05 06 07 08
```

4.3.10 ADC Test

The user can input the analog voltage for PIN1~8 (/dev/adc7 ~ /dev/adc0) on connector J5, the input analog voltage rate is 0v ~ 3.3v, for the 12 bit AD conversion, the system will display the digit voltage.

Input the following commands to check the PIN1 analog voltage:

```
[root@OMAP3EVM /]# adc_test -d /dev/adc7
The channel: /dev/adc0 0x0fff data: 3.2990 V
The channel: /dev/adc0 0x0fff data: 3.2990 V
The channel: /dev/adc0 0x0fff data: 3.2990 V
```

Note:

📖 For pin2, the command is "adc_test -d /dev/adc6". When the pin is empty, the digit voltage is 2.2V.

4.3.11 SD Card Test

1. Connect to the SD card to the SOC8200 board, if the following information appears on the debug port, the SD card has been detected.

```
mmc0: new high speed SD card at address 0001
mmcblk0: mmc0:0001 APPSD 1.85 GiB
 mmcblk0: p1
```

2. Input the following commands to mount the SD card.

```
[root@OMAP3EVM /]# mount -t vfat /dev/mmcblk0p1 /mnt
[root@OMAP3EVM /]# ls /mnt
Windows Embedded CE 6.0 R2    NCP   linux-2.6.24
```

3. Umount the SD card.

```
 [root@OMAP3EVM /]# umount /mnt
```

4.3.12 Buzzer Test

1. Enable the buzzer:

```
[root@OMAP3EVM /]# echo 1 > /sys/class/misc/beep/val
```

2. Disable the buzzer:

```
[root@OMAP3EVM /]# echo 0 > /sys/class/misc/beep/val
```

4.3.13 Audio Test

The board has an audio input and output interface, and we have alsa-utils audio test tools in the filesystem, users can enter the following commands for a test:

1. Recording Test：

```
[root@OMAP3EVM /]# arecord -t wav -c 2 -r 44100 -f S16_LE -v k
Recording WAVE 'k' : Signed 16 bit Little Endian, Rate 44100 Hz, Stereo
Plug PCM: Hardware PCM card 0 'omap3evm' device 0 subdevice 0
Its setup is:
  stream      : CAPTURE
  access      : RW_INTERLEAVED
  format      : S16_LE
  subformat   : STD
  channels    : 2
  rate        : 44100
  exact rate  : 44100 (44100/1)
  msbits      : 16
  buffer_size : 22052
  period_size : 5513
  period_time : 125011
  tstamp_mode : NONE
  period_step : 1
  avail_min   : 5513
  period_event : 0
  start_threshold : 1
  stop_threshold  : 22052
  silence_threshold: 0
  silence_size : 0
  boundary    : 1445199872
  appl_ptr    : 0
  hw_ptr      : 0
```

Note:

Press CONTROL+C to quit the test.

2. Playback Testing :

```
[root@OMAP3EVM /]# aplay -t wav -c 2 -r 44100 -f S16_LE -v k
Playing WAVE 'k' : Signed 16 bit Little Endian, Rate 44100 Hz, Stereo
Plug PCM: Hardware PCM card 0 'omap3evm' device 0 subdevice 0
Its setup is:
  stream      : PLAYBACK
  access      : RW_INTERLEAVED
  format      : S16_LE
  subformat   : STD
  channels    : 2
  rate        : 44100
  exact rate  : 44100 (44100/1)
```

```
msbits      : 16
buffer_size  : 22052
period_size  : 5513
period_time  : 125011
tstamp_mode  : NONE
period_step  : 1
avail_min    : 5513
period_event : 0
start_threshold  : 22052
stop_threshold   : 22052
silence_threshold: 0
silence_size : 0
boundary     : 1445199872
appl_ptr     : 0
hw_ptr       : 0
```

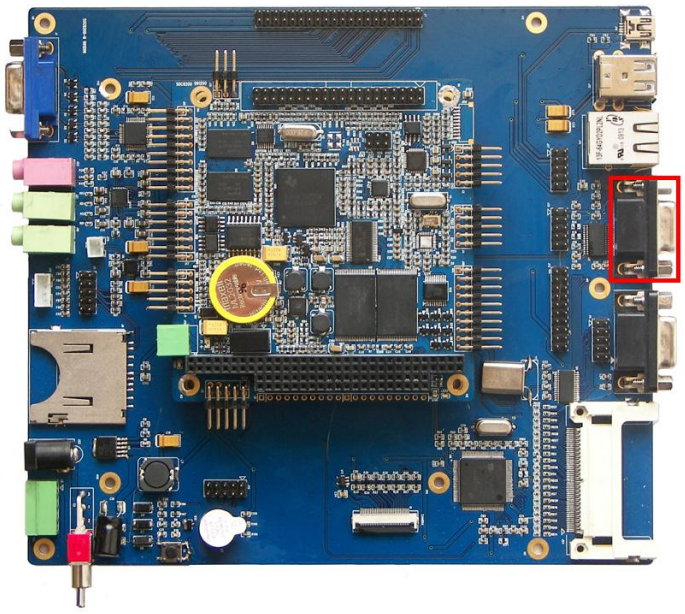
4.3.14 Full-function Serial Port Test

The SOC8200-M has 3 serial ports:

Interface	Type	Encapsulation	Device nodes	dBm	Test program	
Extended serial port 0	Full-function serial port	Interface for DB9	/dev/ttySCMA0	RS232	3 line	com_norts
					9 line	com_rts
Extended serial port 1	Full-function serial port	Interface for IDC	/dev/ttySCMA1	TTL	3 line	com_norts
					9 line	com_rts
Debug port	Three-wire serial port	Interface for IDC	/dev/ttyS2	RS232	System Integration	

1. Test preparation

In the following figure the red box is serial port 0, the test is applied to extended serial port 0.



The steps for connection:

The serial settings for the PC:

- Baud rate: 115200
- Data bit: 8
- Stop bit: 1
- Parity bit: None
- Control flow: Hardware
- DTR: On
- RTS: On

2. Communication Test:

Connect the extended serial port 0 and PC via 3-wire mode:

Input the following commands, the SOC8200 board will send the data "1234567890" to the PC. If the PC has sent the data to SOC8200 board, the board will receive the data too.

```
[root@OMAP3EVM ]# com_norts -d /dev/ttySCMA0
SEND: 1234567890
RECV: www.armkits.com
RECV: www.armkits.com
SEND: 1234567890
RECV: www.armkits.com
RECV: www.armkits.com
SEND: 1234567890
```

Connect the extended serial port 0 and PC via 9-wire mode:

Input the following commands, the SOC8200 board will send the data "1234567890" to the PC. If the PC has sent the data to the SOC8200 board, the board will receive the data too.

```
[root@OMAP3EVM ]# com_rts -d /dev/ttySCMA0
SEND: 1234567890
RECV: SOC8200
RECV: SOC8200
SEND: 1234567890
RECV: SOC8200
RECV: SOC8200
SEND: 1234567890
RECV: SOC8200
RECV: SOC8200
SEND: 1234567890
```

Note:

📖 For the test program com_rts and com_norts, the user can add the " -s " to change the send content.

4.3.15 Digital Output Test

Pins 1 to 8 on the connector J6 can output the digital voltages 0V and 3.3V, the default output is 3.3v.

1. Device introduction:

```
[root@OMAP3EVM /]# cd /sys/class/misc/digital/
[root@OMAP3EVM digital]# ls
dev       out2      out4      out6      out8      power     uevent
out1      out3      out5      out7      outall    subsystem
```

As above, out1 corresponds to the pin1; the outall corresponds to pins 1 to 8.

2. Set only one pin voltage (pin3):

```
[root@OMAP3EVM digital]# echo 1 > out3
[root@OMAP3EVM digital]# cat out3
1
```

3. Set all pins voltage:

```
 [root@OMAP3EVM digital]# echo aa > outall
[root@OMAP3EVM digital]# cat outall
aa
```

4.4 Linux Image Update

The SOC8200 supports MMC/SD boot or NAND boot; different start-up modes have different methods for updating the image. Here we will introduce the image update process under different start-up modes.

4.4.1 Update the image on an SD card

Preparation

The first step is to format the SD card. The HP USB Disk Storage Format Tool is recommended:

The software can be downloaded from:

http://www.embest-tech.com/resource/download/HP-USB-Disk-Storage-Format-Tool.rar

- Insert the TF card into the card reader of the PC.
- Open the HP USB Disk Storage Format Tool, the following window will be displayed.

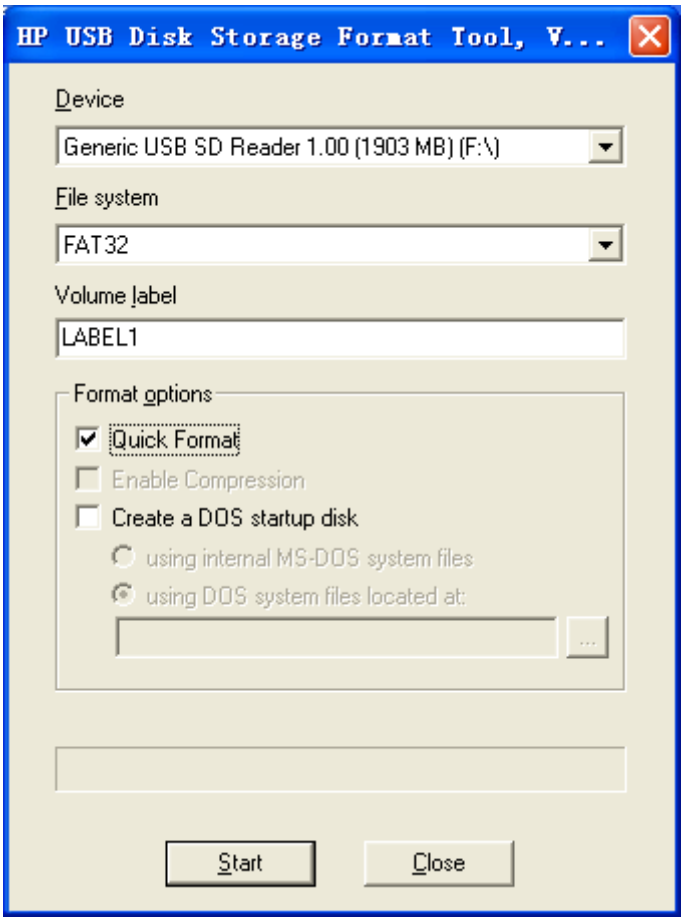


Figure 9: HP USB Disk Storage Format Tool Settings

- Select "FAT32" from the file system drop down box.
- Click "Start".
- When formatting is completed, click "OK".

Note:

🕮 HP USB Disk Storage Format Tool will erase the partitions of TF card.

🕮 Use other format tool may cause the failure of the TF card booting.

To prepare the SD card file:

- Copy the all the files from the directory: disk/linux/image to the SD card.
- Depending on your display device LCD (4.3” or 7”) or VGA, rename “uImage_xx” as uImage

Update the image

1. Enter u-boot

Insert the SD card into the SOC8200 board, and make sure you have connected the jumper cap on J24 (see Figure 10)

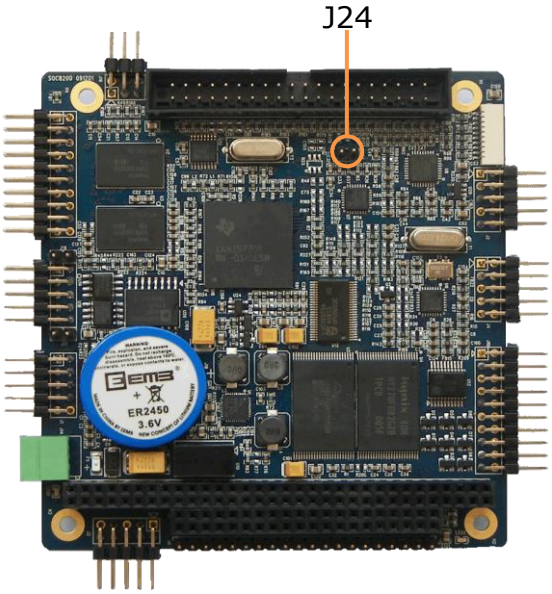


Figure 10: J24 Location

Note:

🕮 Regardless of whether you are updating the image for the SD card or NAND Flash, you must start the image from the SD card.

The user has to enter u-boot command line mode first:

```
40X

Texas Instruments X-Loader 1.45 (Mar 19 2010 - 16:09:58)
Starting X-loader on MMC
Reading boot sector

213544 Bytes Read from MMC
Starting OS Bootloader from MMC...
Starting OS Bootloader...



U-Boot 2009.11-svn ( 3鏈?19 2010 - 16:14:31)

OMAP34xx/35xx-GP ES1.0, CPU-OPP2 L3-165MHz
AM3517EVM Board + LPDDR/NAND
I2C:   ready
DRAM:  256 MB
NAND:  256 MiB
In:    serial
Out:   serial
Err:   serial
Die ID #7964000000000000001543b2106011005
Net:   davinci_emac_initialize
Ethernet PHY: GENERIC @ 0x00
DaVinci EMAC
Hit any key to stop autoboot:  3
```

At this point the system will count down for 3 seconds. Then press any key to enter u-boot command line mode.

2. Set U-boot parameters

Input the following (input characters in bold) into u-boot.

Set the boot baud rate, boot from ram0 (SD card) and select the file system as ext2:

```
OMAP3517EVM # setenv bootargs console=ttyS2, 115200n8 root=/dev/ram0
rw rootfstype=ext2 initrd=0x81600000, 40M
```

Set the image (uImage, ramdisk.gz) boot.from SD card:

```
OMAP3517EVM # setenv bootcmd 'mmc init\; fatload mmc 0 80300000 uImage\;
fatload mmc 0 81600000 ramdisk.gz\; bootm 0x80300000'
```

Save the env and boot the kernel:

```
OMAP3517EVM # saveenv

OMAP3517EVM # boot
```

4.4.2 Update the image in the NAND Flash

Preperation

Refer to NAND Flash Boot on p27

Update system image

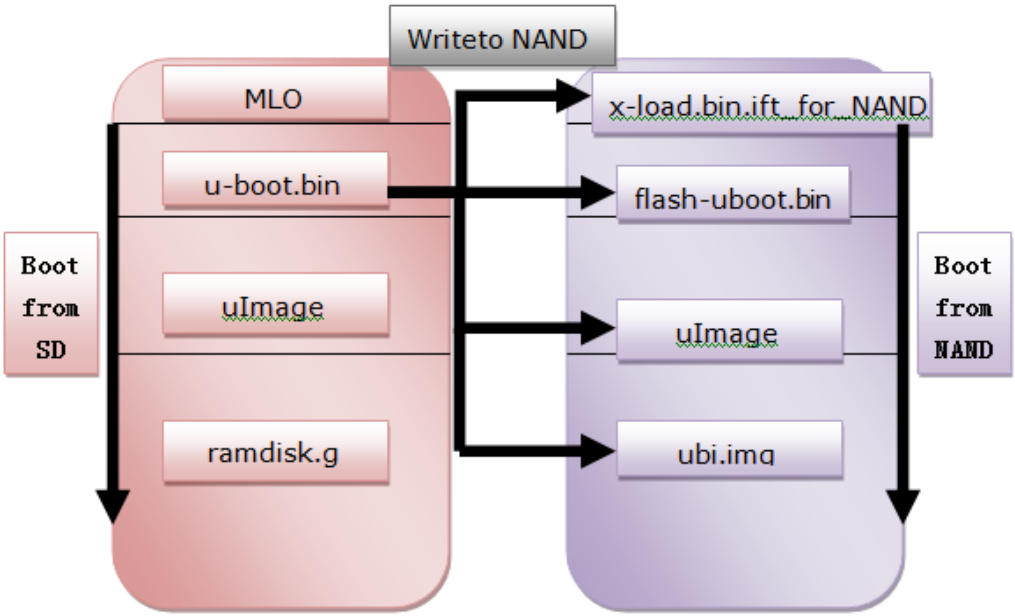


Figure 11: NAND Flash Update Principles

To update the image in the NAND Flash you must input the commands into the u-boot shell, please refer to the following steps:

Updating the x-loader boot image

Input the following commands into the u-boot shell to update images on the SD card:

```
mmc init
fatload mmc 0 80000000 x-load.bin.ift_for_NAND
nand erase 0 80000
nandecc hw
nand write.i 80000000 0 $filesize
```

Updating the u-boot boot image

Input the following commands into the u-boot shell to update images on the SD card:

```
mmc init
fatload mmc 0 80000000 flash-uboot.bin
nand erase 80000 160000
nandecc sw
nand write.i 80000000 80000 $filesize
```

Updating the kernel boot image

Input the following commands into the u-boot shell to update images on the SD card:

```
mmc init
fatload mmc 0 80000000 uImage
nand erase 280000 300000
nandecc sw
nand write.i 80000000 280000 $filesize
```

Updating the filesystem boot image

Input the following commands into the u-boot shell to update images on the SD card:

```
mmc init
fatload mmc 0 81000000 ubi.img
nand erase 680000
nandecc sw
nand write.i 81000000 680000 $filesize
```

Modify the u-boot environment parameters

Input the following commands into the u-boot shell:

```
OMAP3517EVM # setenv bootargs console=ttyS2, 115200n8 ubi.mtd=4 root=ubi0: rootfs
rootfstype=ubifs
OMAP3517EVM # setenv bootcmd nand read.i 80300000 280000 300000\; bootm 80300000
OMAP3517EVM # saveenv
```

4.5 Linux System Deveplopment

This section will introduce how to establish a Linux system development platform to run on the SOC8200 hardware platform with the use of the SOC8200 BSP. Details to be provided contain the formation of a cross compilation environment, the generation of a system image and a demonstration of how to customize the system.

For the SD card, after formatting and dividing into FAT and EXT3 under an ubuntu system, the FAT partition needs reformatting under a windows system, otherwise, start-up with SD card can not be done.

4.5.1 Install the cross compilation environment

The user must form an arm Linux cross compilation environment before developing for the SOC8200. We will use the ubuntu operating system as an example to introduce the formation of a cross compilation environment. The operation in Linux is similar with that in ubuntu system.

Insert the CD, ubuntu will mount the CD under the /media/cdrom directory, and the cross compilation tool will be put under the /media/cdrom/linux/tools directory.

Users can execute the following commands to start up the installation of the cross compilation tool:

```
cd /media/cdrom/linux/tools
tar xvjf arm-2009q1-203-arm-none-linux-gnueabi-i686-pc-linux-gnu.tar.bz2 -C
/home/embest
```

Note:

📖 This manual uses /home/embest as the default installation directory. This can be changed if required.

4.5.2 The installation of other tools

Other tools included in linux/tools directory of the CD may be used for source code. Users can execute the following commands for installation:

```
mkdir /home/embest/tools
cp /media/cdrom/linux/tools/mkimage /home/embest/tools
cp /media/cdrom/linux/tools/signGP /home/embest/tools
```

```
cp /media/cdrom/linux/tools/mkfs.ubifs /home/embest/tools
cp /media/cdrom/linux/tools/ubinize /home/embest/tools
cp /media/cdrom/linux/tools/ ubinize.cfg /home/embest/tools
```

4.5.3 Adding environment variable

After installation of the above tools, those tools can be added into environment variable with the following command:

```
export PATH=/home/embest/arm-2009q1/bin:/home/embest/tools:$PATH
```

Note:

📖 Users can add this to the .barsrc file, allowing the environment variable to be finished as the system starts.

4.5.4 System complation

Preparation

The source code of each part of the system is under the linux/source directory of the CD. Users can copy it to the system and unzip it before developing. For example:

```
mkdir /home/embest/work
cd /home/embest/work
tar xvf /media/cdrom/linux/source/ x-loader-03.00.00.04.tar.bz2
tar xvf /media/cdrom/linux/source/ u-boot-03.00.00.04.tar.bz2
tar xvf /media/cdrom/linux/source/ linux-03.00.00.04.tar.bz2
sudo tar xvf /media/cdrom/linux/source/rootfs.tar.bz2
```

When the above steps are finished, the current directory will contain the three directories: linux-2.6.22-omap, u-boot-1.3.3 and x-load-1.41.

x-loader image generation

The DevKit8200 supports MMC/SD boot or NAND boot. The burned x-loader image files are different for each different boot mode, and the corresponding methods for mapping will differ too.

We will introduce the generation of x-loader image files under the different boot modes.

1. To generate the x-loader image file MLO used for SD card start-up

```
cd x-load-03.00.00.04
make distclean
make am3517evm_config
make
signGP x-load.bin
mv x-load.bin.ift MLO
```

When the above steps are finished, the current directory will contain the required MLO file.

2. To generate the x-load.bin.ift_for_NAND start-up

To alter the file x-loader-1.4.1/include/configs/am3517evm.h and annotate the following:

```
vi x-loader-03.00.00.04/include/configs/am3517evm.h
//#define CONFIG_MMC            1
```

Cross compilation

```
cd x-load-1.41
make distclean
make am3517evm _config
make
signGP x-load.bin
mv x-load.bin.ift x-load.bin.ift_for_NAND
```

When the above steps are finished, the current directory will contain the required x-load.bin.ift_for_NAND file.

u-boot image generation

```
cd u-boot-03.00.00.04/
make distclean
make am3517_evm_config
make
```

When the above steps are finished, the current directory will contain the required u-boot.bin file.

Kernel image generation

The user may change linux-03.00.00.04/drivers/video/omap2/displays/

panel-sharp-lq043t1dg01.c, the default display is VGA.

```
//#define LCD_43inch    1
//#define LCD_7inch 1
#define VGA     1
```

Compilation

```
cd linux-03.00.00.04/
make distclean
cp arch/arm/configs/omap3_soc8200_defconfig .config
make
make uImage
```

When the above steps are finished, the arch/arm/boot directory will contain the required uImage file.

ubifs image generation

```
cd /home/embest/work
sudo /home/embest/tools/mkfs.ubifs -r rootfs -m 2048 -e 129024 -c 812 -o ubifs.img
sudo /home/embest/tools/ubinize -o ubi.img -m 2048 -p 128KiB -s 512 /home/embest/tools/ubinize.cfg
```

When the above steps are finished, the current directory will contain the required ubi.img file.

4.6 Linux System Customization

The Linux kernel has many configuration options. From the default configuration, users can add or delete some configurations to suit different needs. The following example illustrates the general process of system customization.

4.6.1 4 Alteration of kernel configuration

The kernel source code provides the default configuration file:

arch/arm/configs/omap3_soc8200_defconfig

Users can customize the system on the basis of this file

```
cd linux-03.00.00.04/
cp arch/arm/configs/omap3_soc8200_defconfig .config
make menuconfig
```

The example that we use to introduce the system customization is a usb gadget simulating a usb mass storage device:

1. Select Device drivers

Symbol: USB_FILE_STORAGE [=m]

Prompt: File-backed Storage Gadget

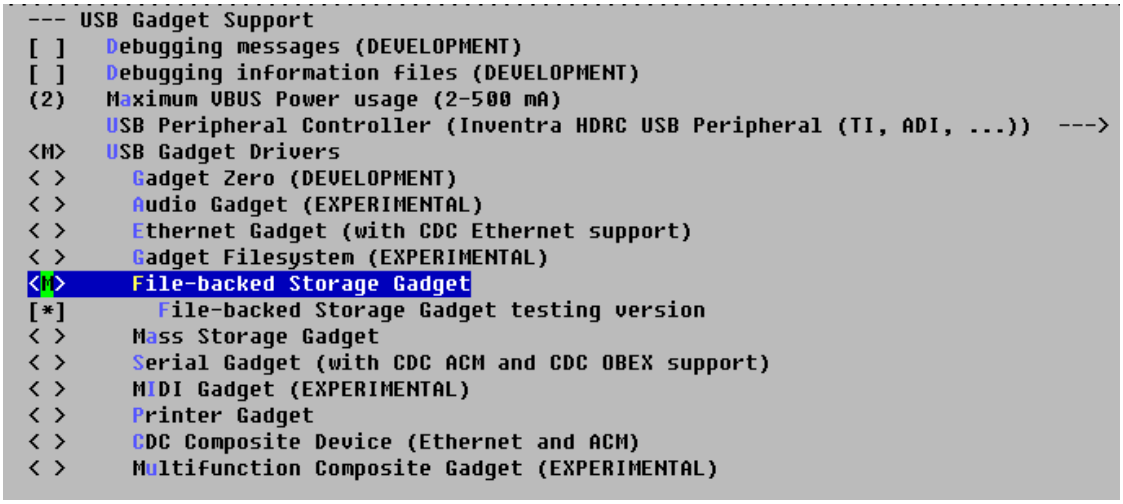
Defined at: drivers/usb/gadget/Kconfig:713

Depends on: <choice> && BLOCK [=y]

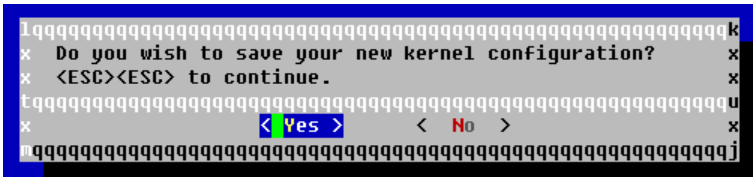
Location:

- Device Drivers
- USB support (USB_SUPPORT [=y])
- USB Gadget Support (USB_GADGET [=y])
- USB Gadget Drivers (<choice> [=m])

2. Select the following option: (File-backed Storage Gadget).


--- USB Gadget Support
[ ] Debugging messages (DEVELOPMENT)
[ ] Debugging information files (DEVELOPMENT)
(2) Maximum VBUS Power usage (2-500 mA)
USB Peripheral Controller (Inventra HDRC USB Peripheral (TI, ADI, ...)) --->
<M> USB Gadget Drivers
< > Gadget Zero (DEVELOPMENT)
< > Audio Gadget (EXPERIMENTAL)
< > Ethernet Gadget (with CDC Ethernet support)
< > Gadget Filesystem (EXPERIMENTAL)
<M> File-backed Storage Gadget
[*] File-backed Storage Gadget testing version
< > Mass Storage Gadget
< > Serial Gadget (with CDC ACM and CDC OBEX support)
< > MIDI Gadget (EXPERIMENTAL)
< > Printer Gadget
< > CDC Composite Device (Ethernet and ACM)
< > Multifunction Composite Gadget (EXPERIMENTAL)

3. Select “exit” until the following is displayed.



Select ”Yes”.

4.6.2 Compilation

Save the configuration and execute the following command to recompile the kernel:

```
make
make uImage
```

After the above steps are finished, the arch/arm/boot directory will contain a new kernel image uImage; and drivers/usb/gadget directory will contain a new module file **g_file_storage.ko**.

4.6.3 Test

Update the kernel image file ulmage on the SD card, copy the file g_file_storage.ko to the SD card and reboot the system from the SD. Execute the following commands to stimulate the SOC8200 into acting as a USB mass storage device for PC's visit:

```
root@DevKit8000:~# mount –t vfat /dev/mmcblk0p1 /mnt
root@DevKit8000:~# cd /mnt
root@DevKit8000:/mnt# insmod g_file_storage.ko file=/dev/mmcblk0p1 stall=0
removable=1
g_file_storage gadget: File-backed Storage Gadget, version: 7 August 2007
g_file_storage gadget: Number of LUNs=1
g_file_storage gadget-lun0: ro=0, file: /dev/mmcblk0p1
musb_hdrc musb_hdrc: MUSB HDRC host driver
musb_hdrc musb_hdrc: new USB bus registered, assigned bus number 2
usb usb2: configuration #1 chosen from 1 choice
hub 2-0:1.0: USB hub found
hub 2-0:1.0: 1 port detected
```

Use the USB cable (USB mini B to USB A) to connect the development board to the PC, the PC will notify you that a USB mass storage device has been found.

Note:

📖 Please make sure that the kernel image has been updated, otherwise, the module: g_file_storage.ko will fail to load and a message similar to the following will be displayed: `insmod: cannot insert '/media/mmcblk0p1/g_file_storage.ko': Device or resource busy`

4.7 Linux Application Development

This section will introduce how to conduct application development on the SOC8200 hardware platform, including the formation of an SOC8200 software environment. Examples will be taken to show the general process of the development of SOC8200 applications.

4.7.1 LED application development

1. Coding

Led_acc.c source code is shown below, The led lamps on the development board will flash.

```
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/ioctl.h>
#include <fcntl.h>

#define LED "/sys/class/leds/led/brightness"

int main(int argc, char *argv[])
{
        int f_led;
        unsigned char i = 0;
        unsigned char dat;
        if((f_led = open(LED, O_RDWR)) < 0){
            printf("error in open %s",LED);
            return -1;
        }

    for(;;){
            i++;
            //dat = i&0x1 ? '1':'0';
            //dat = (i&0x2)>>1 ? '1':'0';
            dat = (i&0x4)>>2 ? '1':'0';
            write(f_led, &dat, sizeof(dat));

        usleep(300000);
        }
}
```

2. Cross compilation

```
arm-none-linux-gnueabi-gcc led_acc.c -o led_acc
```

3. Download and run

Resources can be put into the SOC8200 board system via SD card or U flash card or download. Then enter the directory containing the file led_acc exists, and input the following commands and enter, then the led_acc will run in the background.

```
./led_acc &
```

5 WinCE System

5.1 WinCE System Overview

The SOC8200 software system includes: pre-compiled images and applications and their corresponding static library, dynamic link library, header file and source code; cross compilation tools, auxiliary tools for development. Images, applications, Cross compilation tools used for generating image and applications can be downloaded from Microsoft. Image, application, source code and auxiliary tools for the SOC8200 can be found on the release CD or SD card provided with the SOC8200.

The SD card has the following software:

- X-Loader image (MLO)
- Ethernet Bootloader (EBOOT) image (EBOOTSD.nb0)
- Windows Embedded CE 6.0 sample OS image (NK.bin)

The CD includes:

- Windows Embedded CE 6.0 SOC8200 Board Support Package (BSP) source code for TI OMAP35X
- Windows Embedded CE 6.0 project for SOC8200 BSP
- SOC8200 application development example (source code)
- Auxiliary development tools

This section mainly introduces software for the SOC8200 system and covers the description of pre-compiled images and BSP and test kit as well as some functions and features of the various images and applications in the CD.

5.1.1 Pre-compiled image

The pre-compiled images include a boot image X-Loader, EBOOT and a sample OS image. X-Loader is a first level bootloader. After the start-up of system, the ROM inside the CPU will copy the x-loader to internal RAM and perform work. Its main function is to initialize the CPU, and copy EBOOT to the DDR memory and execute EBOOT. EBOOT is a second level bootloader, by default, it will copy system image to the DDR memory and hand the

control to the operating system. EBOOT also can provide related functions to manage the basic hardware and set the shared data in the operating system.

Windows Embedded CE 6.0 provides multimedia modules, industry modules, PDA modules, mobile modules and micro kernel modules, the user can choose the module that they want.Taking Mobile Handheld as an example, the pre-compiled images support the following:

Image	**Feature**
X-Loader	To boot EBOOT
EBOOT	To boot the operating system from the network (network card or RNDIS)
	To boot the operating system with SD card
	To boot the operating system from the NAND Flash
Demonstrated operating system	Windows Explorer
	Console Window
	CAB File Installer/Uninstaller
	Internet Explorer 6.0
	ActiveSync
	Power Management (Full)
	.NET Compact Framework 3.5
	Hive-based Registry
	RAM and ROM File System
	Device Drivers

5.1.2 Board Support Package (BSP)

The SOC8200 BSP is used to customize the boot image and Windows Embedded CE 6.0 OS image to run on SOC8200 hardware platform. It supports the following:

Module	Feature
X-Loader module	NAND
	ONENAND
	SD
EBOOT module	NAND
	ONENAND
	SD
OAL module	ILT
	REBOOT
	Watchdog
	RTC
KITL module	RNDIS KITL
Driver module	NLED driver
	GPIO/I2C/SPI/MCBSP driver
	Series port driver
	6X6 keyboard driver
	Audio driver
	NAND (K9F1G08) driver
	Display driver (LCD/DVI. S end/TV) /TOUCH driver
	SD/MMC/SDIO driver
	DM9000 network card driver
	USB OTG driver
	USB EHCI driver

	VRFB driver
	DSPLINKK/CMEMK driver
	GPIO keyboard driver
	PWM (TPS65930) driver
	ADC (TPS65930) driver
	ONENAND driver
	SMSC911X network card driver
	CAN driver
	Buzzer driver
Power management module	Backlight driver
	Battery driver
	Sleep / wake-up button driver
	Expansion of power management
Application module	Flash Plug-in and Flash player
	MP3/MPEG4/H264 DSP Hardware decoder
	BSPINFO (control panel)
	CETK

5.2 WinCE System Quick Start

5.2.1 System boot

When you boot the board and operate the system, you may use Hyper terminal software with the following settings:

- Baud rate: 115200
- Data bit: 8
- Parity check: no
- Stop bit: 1
- Flow control: no

5.2.2 Boot from NAND Flash

Please refer to p67

5.2.3 Boot from SD card

Copy the image files **MLO, EBOOTSD.nb0, and NK.bin** from the CD:\WinCE\image\ VGA_1024x768(lcd_800x480 or lcd_480x272)\SD directory to the SD card.

To switch to SD card, you need to short J24.

Note:

📖 The method to update the image from the SD card will be shown in 5.4.1 Updating the image on the SD card p67

5.3 WinCE System Development

5.3.1 Development environment building

Install the cross compilation environment

Developing applications requires the installation and updating of the following software in the order listed:

- Visual Studio 2005
- Visual Studio 2005 SP1
- Visual Studio 2005 SP1 Update for Vista (if applicable)
- ActiveSync 4.5

The development of Windows Embedded CE 6.0 requires the installation and updating of the following software in the order listed:

- Visual Studio 2005
- Visual Studio 2005 SP1
- Visual Studio 2005 SP1 Update for Vista (if applicable)
- Windows Embedded CE 6.0 Platform Builder
- Windows Embedded CE 6.0 SP1
- Windows Embedded CE 6.0 R2
- Windows Embedded CE 6.0 Product Update Rollup 12/31/2008

Note:

🕮 If there is an old CE development environment in the system, the use of the Windows Embedded CE 6.0 development platform may be influenced. Uninstalling the old one and then installing the new one is recommended.

🕮 All software should be installed in the order suggested and to their default paths in order to reduce compatibility errors

5.4 WinCE System complation

If the sample Windows Embedded CE 6.0 OS image in the CD of SOC8200 satisfies your requirements, you just need to add it into your application and get the authorization of the Microsoft Corporation. Otherwise, you will need

to re-customize the system and rebuild the image. This section describes how to use the SOC8200 Board Support Package (BSP) to create the ideal Windows Embedded CE 6.0 system image to run on your SOC8200 hardware platform.

5.4.1 Preparation

Embest Technology Co., Ltd., has integrated drivers and their related resources into the SOC8200. In order to customise the SOC8200s embedded security you need to undertake the following preperation:

Decompress [SOC8200\WinCE\BSP\AM35x_BSP.rar] to obtain the AM35x_BSP directory.

Decompress [SOC8200\WinCE\BSP\COMMON_TI_V1.rar] to obtain the COMMON_TI_V1 directory.

Decompress [SOC8200\WinCE\BSP\AM35x_OSDesign.rar] to obtain the AM35x_OSDesign directory.

- Copy the decompressed directory [SOC8200\WinCE\BSP\AM35x_BSP] to the [C:\WINCE600\PLATFORM] directory.
- Copy the decompressed directory [SOC8200\WinCE\BSP\AM35x_OSDesign] to the [C:\WINCE600\OSDesigns] directory.
- Copy the decompressed directory [SOC8200\WinCE\BSP\COMMON_TI_V1] to the [C:\WINCE600\PLATFORM\COMMON\SRC\SOC\] directory.

Note:

📖 OSDesigns is needed to establish the folder (C:\WINCE600\OSDesigns).

For the 4.3” LCD

Modify C:\wince600\platform\am35x_bsp\src\bsp_common\display\Lcd_cfg.h

```
//-----------------------------------------
//#define lcd_7inch  1
```

```
#define lcd_43inch  1
//#define lcd_vga_1024x768  1
//-----------------------------------------
```

For the 7” LCD

Modify
C:\wince600\platform\am35x_bsp\src\bsp_common\display\Lcd_cfg.h

```
//-----------------------------------------
#define lcd_7inch  1
//#define lcd_43inch  1
//#define lcd_vga_1024x768  1
//-----------------------------------------
```

For the VGA

Modify
C:\wince600\platform\am35x_bsp\src\bsp_common\display\Lcd_cfg.h

```
//-----------------------------------------
//#define lcd_7inch   1
//#define lcd_43inch   1
#define lcd_vga_1024x768   1
//-----------------------------------------
```

Note:

- 🕮 If user needs to use the SOC8200 BS to develop a Windows Embedded CE 6.0 operating system, the construction of a Windows Embedded CE 6.0 development platform is required.
- 🕮 Examples in this manual use the default installation path for Windows Embedded CE 6.0 software (C:\WINCE600).

5.4.2 System Compliation

1. Open the file SOC8200 .sln [C:\WINCE600\OSDesigns\SOC8200] or take the following steps to create a new project:

- Open Visual Studio 2005.
- Select on the menu: File [New->Project].
- Select Platform Builder for CE 6.0 as the template type
- Select a file name and open the Windows Embedded CE 6.0 OS Design Wizard

- Set the Embest SOC8200 BSP in the BSP list.
- Continue to finish the Wizard.

2. Select submenu [Build-> Global Build Settings]

- Copy Files to the Release Directory After Build
- Make Run-Time Image After build

3. If KITL is needed, set Enable Kernel Debugger and Enable KITL in the Build Options page [Project-> Properties].

4. Select [Build-> Build Solution] to build the BSP. These operations cover the whole compilation including sysgen operating system's components. After the entire compilation process is completed, the build commands under Solution Explorer window can be used to save the build time.

5. Images including NK.bin, EBOOTSD.nb0 and MLO and so on will be generated;

6. Copy the files MLO, EBOOTSD.nb0 and NK.bin from [C:\WINCE600\OSDesigns\SOC8200\SOC8200\RelDir\SOC8200_ARMV4I_Release] to the SD card. Insert the SD card into the device and boot the device to test.

5.4.3 System Customization

Windows Embedded CE 6.0 consists of a number of independent modules. Each module provides specific functions, of which some modules can be divided into several components. Each component has a specific feature, allowing the OEM/ODM to customize a stable and efficient version according to their specific application.

Taking Mobile Handheld as a template, sample SOC8200 OS image adds features of components including:

Component	Path
CAB File Installer/Uninstaller	Core OS->CEBASE->Application – End User
.NET Compact Framework 3.5	Core OS->CEBASE->Applications and Services Development->.NET Compact

Component	Path
	Framework 3.5
OS Dependencies for .NET Compact Framework 3.5	Core OS->CEBASE->Applications and Services Development->.NET Compact Framework 3.5-> OS Dependencies for .NET Compact Framework 3.5
Point-to-Point Protocol over Ethernet (PPPoE)	Core OS->CEBASE->Communication Services and Networking->Networking – Wide Area Network (WAN)
USB Function Driver	Core OS->CEBASE->Core OS Services->USB Host Support
USB Host Support	Core OS->CEBASE->Core OS Services->USB Host Support
USB Human Input Device (HID) Class Driver	Core OS->CEBASE->Core OS Services->USB Host Support
USB HID Keyboard and Mouse	Core OS->CEBASE->Core OS Services->USB Host Support-> USB Human Input Device (HID) Class Driver
USB Storage Class Driver	Core OS->CEBASE->Core OS Services->USB Host Support
RAM and ROM File System	Core OS->CEBASE->File Systems and Data Store->File System – Internal (Choose 1)
Hive-based Registry	Core OS->CEBASE->File Systems and Data Store->Registry Storage – Internal (Choose 1)
exFAT File System	Core OS->CEBASE->File Systems and Data Store->Storage Manager
FAT File System	Core OS->CEBASE->File Systems and Data Store->Storage Manager
Storage Manager Control Panel Applet	Core OS->CEBASE->File Systems and Data Store->Storage Manager
Transaction-Safe FAT File System (TFAT)	Core OS->CEBASE->File Systems and Data Store->Storage Manager

Component	Path
Video/Image Compression Manager	Core OS->CEBASE->Graphics and Multimedia Technologies->Media->Video Codecs and Renderers
Console Window	Core OS->CEBASE->Shell and User Interface->Shell->Command Shell
SD Memory	Device Drivers->SDIO->SDIO Memory
serial	Device Drivers->USB Function->USB Function Clients
Windows Embedded CE Test Kit	Device Drivers

Note:

🕮 Components can be added or deleted in the window Catalog Items View of the Visual Studio 2005 (VS2005) integrated development environment.

5.5 WinCE image update

5.5.1 Updating the image on the SD card

Prepare

Run the HP Disk Storage Format Tool and format the SD card with a FAT or FAT32 filesystem.

Image update

Copy the CD directory WinCE_6\Image\VGA_1024X768(lcd_800x480 or lcd_480x272)\SDdirectory files **MLO、EBOOTSD.nb0、NK.bin** to the SD card.

Note:

🕮 VGA_1024x768 outputs VGA at 1280X768 resolution, lcd_800x480 outputs at 800x480 resolution and lcd_480x272 outputs at 480X272 resolution.

5.5.2 Updating the image in NAND Flash

Prepare

1. Run the HP Disk Storage Format Tool and format the SD card to a FAT or FAT32 filesystem.

2. Copy the image files:

- MLO
- XLDRNAND.nb0
- EBOOTSD.nb0
- NK.bin

from the CD :\winCE\image\ VGA_1024x768 (lcd_800x480 or lcd_480x272) \NAND directory to the SD card.

Image update

1. Set J24 to boot from SD card and insert the SD card then restart your system. HyperTerminal will start printing the output information, at the same time press [SPACE] to enter the EBOOT menu.
2. Press [5] to enter the Flash manage menu.
3. Press [a], [b], [c] separately to write the image (XLDR, EBOOT, NK) to flash.
4. Press [0] to return to the main menu, and press [2], [4], [7], [y] to change the boot device.
5. Remove the SD card. Reboot the system, now the board will boot from the NAND flash.

5.6 Application development

This section introduces how to develop an application running on the SOC8200 hardware platform based on the Windows Embedded CE 6.0 operating system. The following preparations should be made:

Note:

🕮 If user needs to use the SOC8200 BSP to develop a Windows Embedded CE 6.0 operating system, the construction of a Windows Embedded CE 6.0 development platform is required.

- The installation of the Windows Mobile 6 Professional SDK is advised. You can obtain this software from the Microsoft Download Center at: [http://www.microsoft.com/downloads/details.aspx?familyid=06111A3A-A651-4745-88EF-3D48091A390B&displaylang=en].
- The development example in this manual is based on the development of the Windows Mobile 6 Professional SDK.

5.6.1 API and Application Demonstrations

The Application Programming Interface (API) used for SOC8200 application development employs the standard application interface of Windows Embedded CE 6.0. The SOC8200 has an additional GPIO interface based on standard API.

Note:

- For interface definitions of Windows Embedded CE 6.0 standard applications, please refer to related help documents for the MSDN Windows Embedded CE 6.0 API.
- Some interfaces are only used by drivers. They can't be used by applications.

5.6.2 The definition and demonstration of GPIO interface

GPIO device name L"GIO1:" to expand DeviceIoControl interface definition, corresponding IOCTL code includes:

IOCTL Code	Description
IOCTL_GPIO_SETBIT	Set GPIO pin as 1
IOCTL_GPIO_CLRBIT	Set GPIO pin as 0
IOCTL_GPIO_GETBIT	Read GPIO pin
IOCTL_GPIO_SETMODE	Set the working mode of GPIO pin
IOCTL_GPIO_GETMODE	Read the working mode of GPIO pin
IOCTL_GPIO_GETIRQ	Read the corresponding IRQ of GPIO pin

An operation example is shown below:

1. Open GPIO device

```
    HANDLE  hFile  =  CreateFile(_T("GIO1:"),  (GENERIC_READ|GENERIC_WRITE),
(FILE_SHARE_READ|FILE_SHARE_WRITE), 0, OPEN_EXISTING, 0, 0);
```

2. Set/read the working mode of GPIO

```
    DWORD id = 0, mode = 0;
```

Set the working mode of GPIO:

```
    DWORD pInBuffer[2];
```

```
    pInBuffer[0] = id;
    pInBuffer[1] = mode;
    DeviceIoControl(hFile, IOCTL_GPIO_SETMODE, pInBuffer, sizeof(pInBuffer),
NULL, 0, NULL, NULL);
```

Read the working mode of the GPIO:

```
    DeviceIoControl(hFile, IOCTL_GPIO_GETMODE, &id, sizeof(DWORD), &mode,
sizeof(DWORD), NULL, NULL);
```

"id" is the GPIO Pin number, "mode" is GPIO mode, including:

Mode definition	**Description**
GPIO_DIR_OUTPUT	Output mode
GPIO_DIR_INPUT	Input mode
GPIO_INT_LOW_HIGH	Rising edge trigger mode
GPIO_INT_HIGH_LOW	Falling edge trigger mode
GPIO_INT_LOW	low level trigger mode
GPIO_INT_HIGH	high level trigger mode
GPIO_DEBOUNCE_ENABLE	Jumping trigger enable

3. The operation of GPIO Pin

```
    DWORD id = 0, pin = 0;
```

Output high level:

```
    DeviceIoControl(hFile, IOCTL_GPIO_SETBIT, &id, sizeof(DWORD), NULL, 0, NULL,
NULL);
```

Output low level:

```
    DeviceIoControl(hFile, IOCTL_GPIO_CLRBIT, &id, sizeof(DWORD), NULL, 0, NULL,
NULL);
```

Read the pin state

```
    DeviceIoControl(hFile, IOCTL_GPIO_GETBIT, &id, sizeof(DWORD), &pin,
sizeof(DWORD), NULL, NULL);
```

Where "id" is GPIO pin number, "pin" returns to pin state

4. Other optional operations

Read the corresponding IRQ number of GPIO pin

```
    DWORD id = 0, irq = 0;
```

```
    DeviceIoControl(hFile,  IOCTL_GPIO_GETIRQ,  &id,  sizeof(DWORD),  &irq,
sizeof(DWORD), NULL, NULL);
```

Where "id" is GPIO pin number, "irq" returns IRQ number

5. Close GPIO device

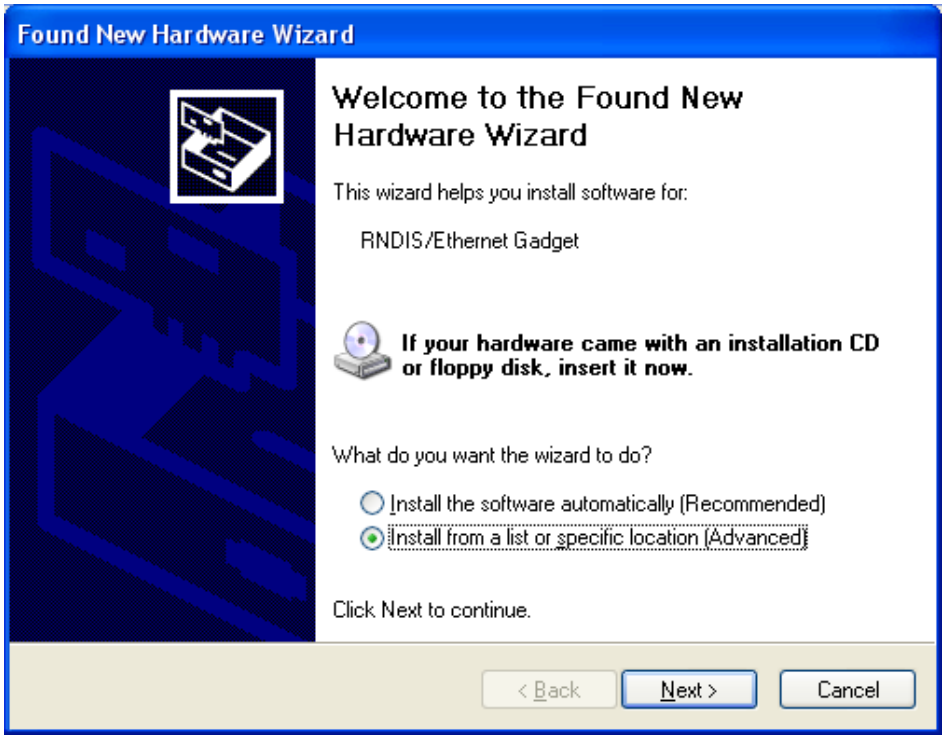
```
    CloseHandle(hFile);
```

Note:

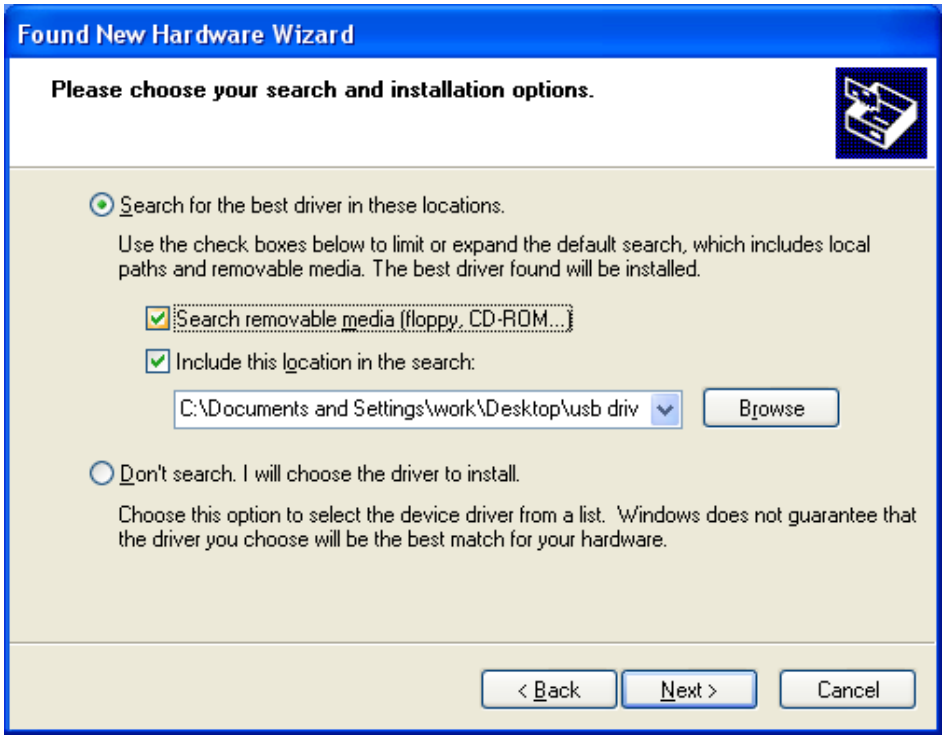
- GPIO pin definition: 0~191 MPU Bank1~6 GPIO pin, 192~209 TPS65930 GPIO 0~17.
- GPIO interrupt mode is used for drivers, applications cannot set this mode.
- For definition of the IOCTL code and GPIO modes, please refer to the CD file [\wince_6\inc\gpio.h] User should include the header file.

Appendix 1: Driver Installation Of Linux USB Ethernet/RNDIS Gadget

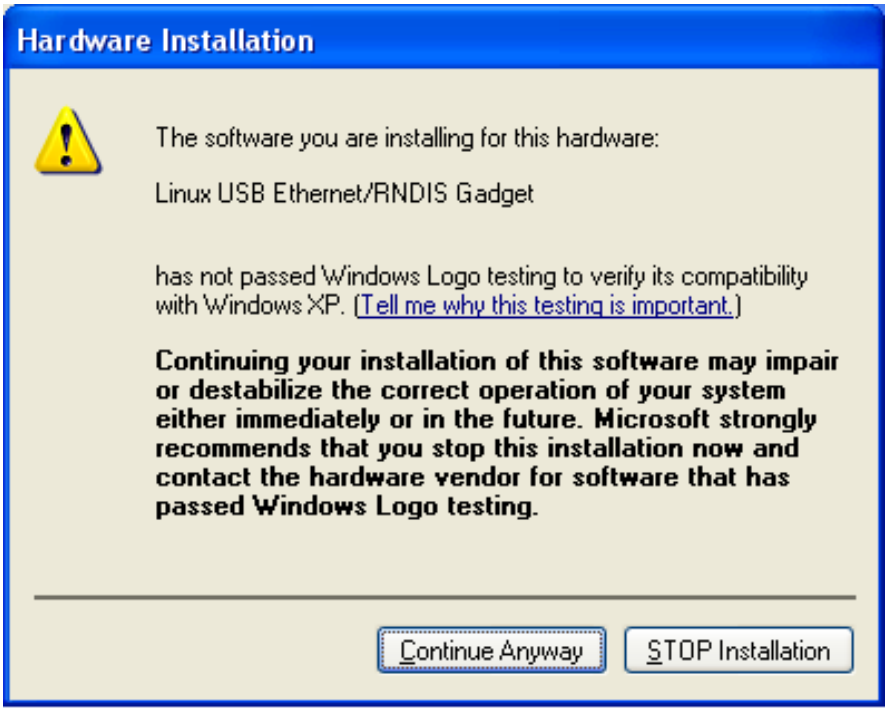
1. If you don't install drivers for the Linux USB Ethernet/RNDIS Gadget, the PC will find the new hardware and give you a hint on the screen, please select "From list or designated location", then click "Next"


Found New Hardware Wizard
Welcome to the Found New Hardware Wizard
This wizard helps you install software for:
RNDIS/Ethernet Gadget
If your hardware came with an installation CD or floppy disk, insert it now.
What do you want the wizard to do?
Install the software automatically (Recommended)
Install from a list or specific location (Advanced)
Click Next to continue.
< Back
Next >
Cancel

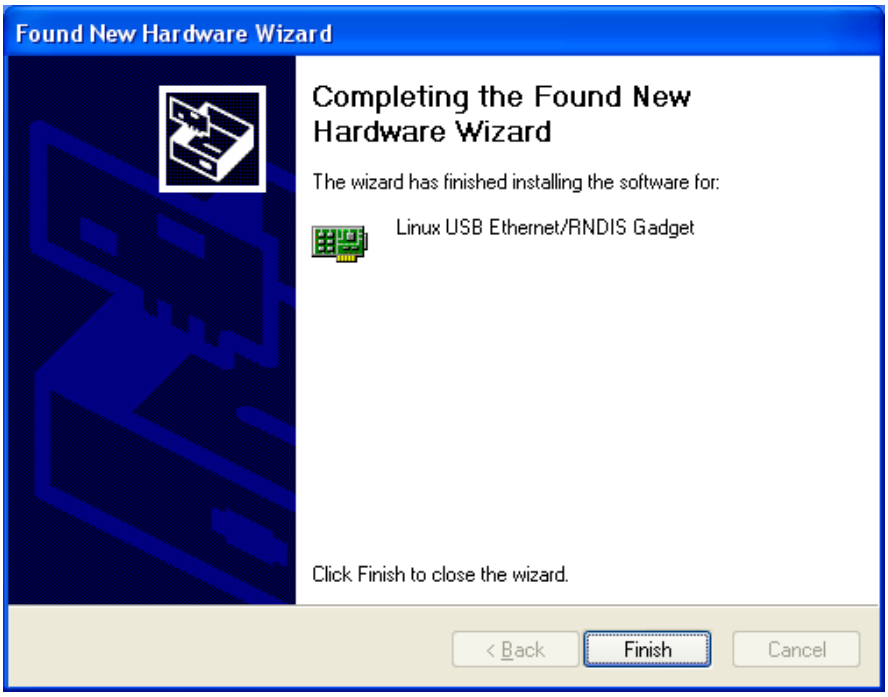
2. Designate the path for the usb driver, the usb driver directory is [disk\linux\tools], then click "Next"


Found New Hardware Wizard
Please choose your search and installation options.
Search for the best driver in these locations.
Use the check boxes below to limit or expand the default search, which includes local paths and removable media. The best driver found will be installed.
Search removable media (floppy, CD-ROM...)
Include this location in the search:
C:\Documents and Settings\work\Desktop\usb driv
Browse
Don't search. I will choose the driver to install.
Choose this option to select the device driver from a list. Windows does not guarantee that the driver you choose will be the best match for your hardware.
< Back
Next >
Cancel

3. When the following appears, select "Continue"


Hardware Installation
The software you are installing for this hardware:
Linux USB Ethernet/RNDIS Gadget
has not passed Windows Logo testing to verify its compatibility with Windows XP. (Tell me why this testing is important.)
Continuing your installation of this software may impair or destabilize the correct operation of your system either immediately or in the future. Microsoft strongly recommends that you stop this installation now and contact the hardware vendor for software that has passed Windows Logo testing.
Continue Anyway
STOP Installation

4. Please wait until the installation is completed


Found New Hardware Wizard
Completing the Found New Hardware Wizard
The wizard has finished installing the software for:
Linux USB Ethernet/RNDIS Gadget
Click Finish to close the wizard.
< Back
Finish
Cancel

Appendix 2: Making a Linux Boot Disk

The following content will show you how to create a dual-partition flash disk for booting up a Linux system from the first partition, while saving the root filesystem in the second one;

1. Insert a TF card into a TF card reader and then connect the reader to your PC; execute the following instruction in an Ubuntu system to view the device name of the TF card;

```
$ dmesg | tail
```

Device Information:

```
...
[ 6854.215650] sd 7:0:0:0: [sdc] Mode Sense: 0b 00 00 08
[ 6854.215653] sd 7:0:0:0: [sdc] Assuming drive cache: write
through
[ 6854.215659]  sdc: sdc1
[ 6854.218079] sd 7:0:0:0: [sdc] Attached SCSI removable disk
[ 6854.218135] sd 7:0:0:0: Attached scsi generic sg2 type 0
...
```

The above information shows the TF card device name is **/dev/sdc**;

2. Execute the following instruction to view the path where Ubuntu has mounted the device automatically;

```
$ df -h
```

Device Path:

```
. Filesystem          Size  Used Avail Use% Mounted on
...
/dev/sdc1           400M   94M  307M  24% /media/disk
```

At the end of the line starting from **/dev/sdc1** you can see the device path is **/media/disk**;

Note:

🕮 If TF card has two or more partitions, there would be multiple paths such as /dev/sdc1, /dev/sdc2 and /dev/sdc3 corresponding to the partitions.

3. Execute the following instruction to unmount the device;

```
$ umount /media/disk
```

4. Execute an **fdisk** instruction;

```
$ sudo fdisk /dev/sdc
```

Please make sure you type the device path for the whole device, not one of the partitions such as /dev/sdc1 or /dev/sdc2;

5. After executing the above instruction, type **p** to print the partition records of the device as shown below;

```
Command (m for help): [ p ]

Disk /dev/sdc: 2021 MB, 2021654528 bytes
255 heads, 63 sectors/track, 245 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
   Device Boot      Start         End      Blocks   Id  System
/dev/sdc1   *           1         246     1974240+   c  W95 FAT32
(LBA)
Partition 1 has different physical/logical endings:
    phys=(244, 254, 63) logical=(245, 200, 19)
```

Write down the total bytes shown in the above information, for example 2021654528 bytes, and then type **d** to delete all the partitions;

6. If you do not find information **255 heads** and **63 sectors/track** in the above table, please go though the following steps to recover the TF card;

7. Type the letters as shown in the following table to set Heads and Sectors;

```
Command (m for help): [ x ] (type x to enter expert mode)
Expert Command (m for help): [ h ] (type h to set heads)
Number of heads (1-256, default xxx): [ 255 ] (set heads to 255)
Expert Command (m for help): [ s ] (type s to set sectors)
Number of sectors (1-63, default xxx): [ 63 ] (set sector to
63)
```

8. Use the following equation to calculate the number of Cylinders;

Cylinders = the total bytes written down previously÷255÷63÷512

Type the letters as shown in the following table to set Cylinders;

```
Expert Command (m for help): [ c ] (type c to set cylinders)
Number of cylinders (1-256, default xxx): (enter the number of
cylinders calculated above)...
Expert Command (m for help): [ r ] (type r to go back to normal
mode)
```

9. Type **p** to check the parameters set just now as shown below;

```
Command (m for help): [ p ]
63 sectors/track, 245 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
```

10. Create a FAT32 partition and transfer files from Windows according to the operations in the follow table;

```
Command (m for help): [ n ] (type n to start creating partition)
Command action
   e   extended
   p   primary partition (1-4)
[ p ] (type p to create primary partition)
Partition number (1-4): [ 1 ] (set the partition number to 1)
First cylinder (1-245, default 1): [ ] (press Enter key on your
keyboard)
Using default value 1
Last cylinder or +size or +sizeM or +sizeK (1-61, default 61):
[ +5 ]  (enter +5)

Command (m for help): [ t ]  (type t)
```

```
Selected partition 1
Hex code (type L to list codes): [ c ] (type c to set partition
type)
Changed system type of partition 1 to c (W95 FAT32 (LBA))
```

11. Type **a** and **1** to set the TF card to bootable mode;

```
Command (m for help): [ a ]
Partition number (1-4): [ 1 ]
```

12. Type the letters as shown in the following table to create a partition for the root filesystem;

```
Command (m for help): [ n ] (type n to create a partition)
Command action
   e   extended
   p   primary partition (1-4)
[ p ] (type p to select primary partition)
Partition number (1-4): [ 2 ] (set partition number to 2)
First cylinder (7-61, default 7): [ ] (press Enter key on your
keyboard)
Using default value 52
Last cylinder or +size or +sizeM or +sizeK (7-61, default 61):
[ ] (press Enter key)
Using default value 245
```

13. Type p to check the created partitions as shown below;

```
Command (m for help): [ p ]

Disk /dev/sdc: 2021 MB, 2021654528 bytes
255 heads, 63 sectors/track, 245 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sdc1   *           1           6      409626    c  W95 FAT32
(LBA)
/dev/sdc2               7          61     1558305   83  Linux
```

14. Type **w** to save new partition records as shown below;

```
Command (m for help): [ w ]
The partition table has been altered!

Calling ioctl() to re-read partition table.

WARNING: Re-reading the partition table failed with error 16:
Device or resource busy.
The kernel still uses the old table.
The new table will be used at the next reboot.

WARNING: If you have created or modified any DOS 6.x
partitions, please see the fdisk manual page for additional
information.
Syncing disks.
```

15. Execute the following instructions to form the new partitions;

```
$ [sudo mkfs.msdos -F 32 /dev/sdc1 -n LABEL1]
$ [sudo mkfs.ext3 -L LABEL2 /dev/sdc2]
```

Note:

- The drive labels LABEL1 and LABEL2 in the above instructions are just for reference, you can use your own labels if required;
- After the FAT and EXT3 partitions are formatted, the FAT partition needs to be formatted again under a Windows system to avoid failure when booting from the TF card.

Appendix 3: TFTP Server Setup

1. Install client

```
$>sudo apt-get install tftp-hpa
$>sudo apt-get install tftpd-hpa
```

2. Install inet

```
$>sudo apt-get install xinetd
$>sudo apt-get install netkit-inetd
```

3. Configure the server

First, create tftpboot under root directory, and set the properties as "a random user can write and read"

```
$>cd /
$>sudo mkdir tftpboot
$>sudo chmod 777 tftpboot
```

Secondly, add in /etc/inetd.conf:

```
$>sudo  vi  /etc/inetd.conf     //copy the follow word to this file
tftpd dgram udp wait root /usr/sbin/in.tftpd /usr/sbin/in.tftpd -s /tftpboot
```

Then, reload inetd process:

```
$>sudo /etc/init.d/inetd reload
```

Finally, enter directory /etc/xinetd.d/, and create a new file tftp and put the designated content into file tftp:

```
$>cd /etc/xinetd.d/
$>sudo touch tftp
$>sudo vi tftp          ////copy the follow word to tftp file
service tftp
{
    disable = no
    socket_type   = dgram
    protocol      = udp
```

```
    wait            = yes
    user            = root
    server          = /usr/sbin/in.tftpd
    server_args     = -s /tftpboot -c
    per_source      = 11
    cps             = 100 2
}
```

4. Reboot the server:

```
$>sudo /etc/init.d/xinetd restart
$>sudo in.tftpd -l /tftpboot
```

5. Test the server

Conduct a test; create a file under folder /tftpboot

```
$>touch abc
```

Enter into another folder

```
$>tftp 192.168.1.15  (192.168.1.15was the server IP)
$>tftp> get abc
```

If the download is successful, this means the server has been installed.

Appendix 4: WinCE related resource links

1. Visual Studio 2005 SP1 Update for Vista (if applicable)

http://download.microsoft.com/download/c/7/d/c7d9b927-f4e6-4ab2-8399-79a2d5cdfac9/VS80sp1-KB932232-X86-ENU.exe

2. Windows Embedded CE 6.0 Platform Builder Service Pack 1

http://www.microsoft.com/downloads/details.aspx?familyid=BF0DC0E3-8575-4860-A8E3-290ADF242678&displaylang=en

3. Windows Embedded CE 6.0 R2

http://www.microsoft.com/downloads/details.aspx?FamilyID=f41fc7c1-f0f4-4fd6-9366-b61e0ab59565&displaylang=en

4. Windows Embedded CE 6.0 R3

http://download.microsoft.com/download/F/5/2/F5296720-250A-4055-991C-0CEA5DE11436/CE6R3.iso

5. WinCEPB60-091231-Product-Update-Rollup-Armv4I.msi

http://download.microsoft.com/download/E/D/7/ED779010-1B2E-4ACA-BF9F-9F1D0EF8052B/WinCEPB60-091231-Product-Update-Rollup-Armv4I.msi

6. Viewers for Windows Embedded CE 6.0 R3

http://download.microsoft.com/download/3/3/8/3383B6CE-F70A-4A2C-873A-8C67D3CF55F6/WesttekFileViewers6.exe

7. Windows Mobile 6 Professional SDK Refresh.msi

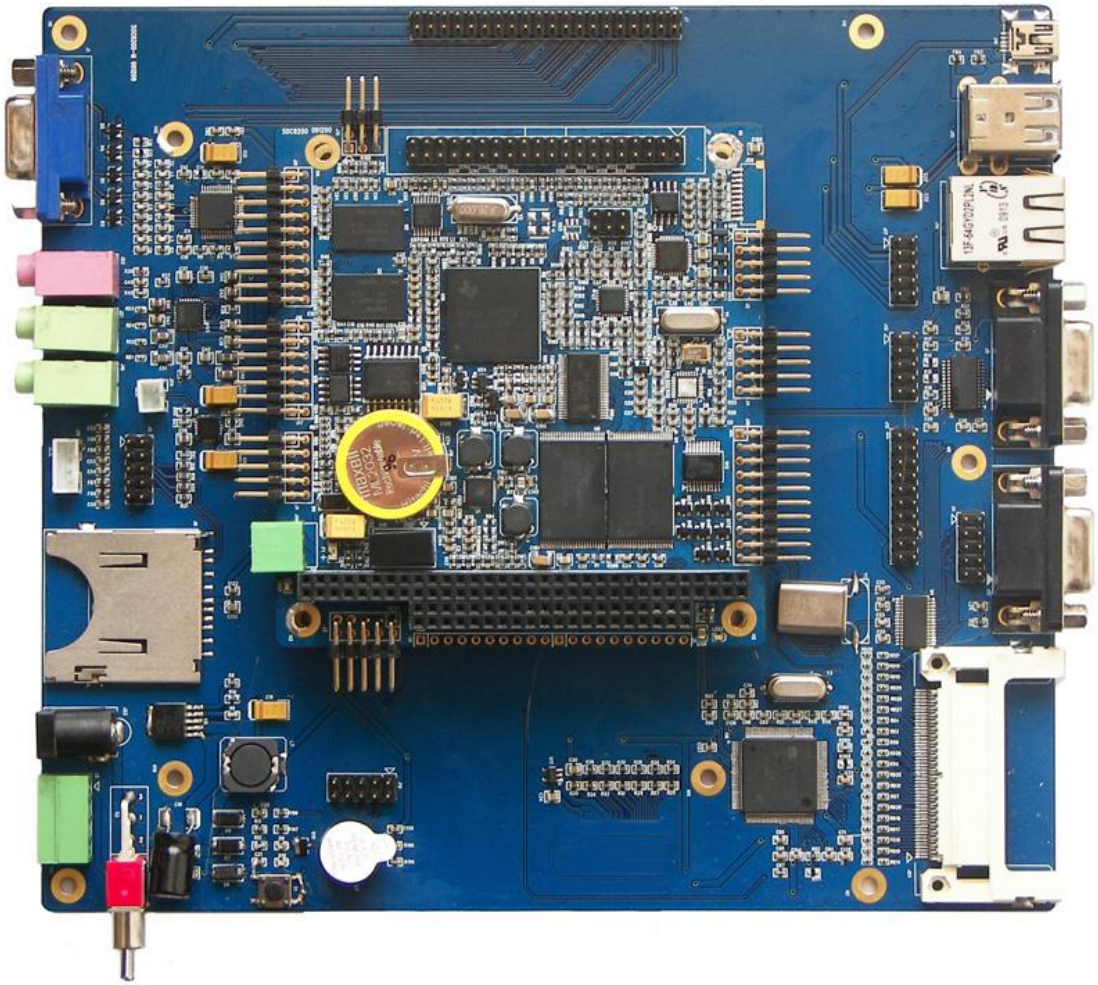
http://download.microsoft.com/download/f/2/3/f232f773-7edc-4300-be07-d3b76a5b3a91/Windows%20Mobile%206%20Professional%20SDK%20Refresh.msi

8. Windows Embedded CE 6.0 USB Camera Driver.msi

http://download.microsoft.com/download/f/a/1/fa1aaef1-6ae3-4cf3-ab95-b01d3e428403/Windows%20Embedded%20CE%206.0%20USB%20Camera%20Driver.msi

Appendix 5: Expansion Board

The customer can evaluate the AM3517 via the SOC8200 expansion board(SOC8200-M), to experience the full functionality of the AM3517 processor. The customer can use both the **single board computer** and the **function Interface board** to add the product functions, thus reducing product development cycles and achieving a faster time to market.



5.1 SOC8200-M Evaluation Suite

Audio/video interface:

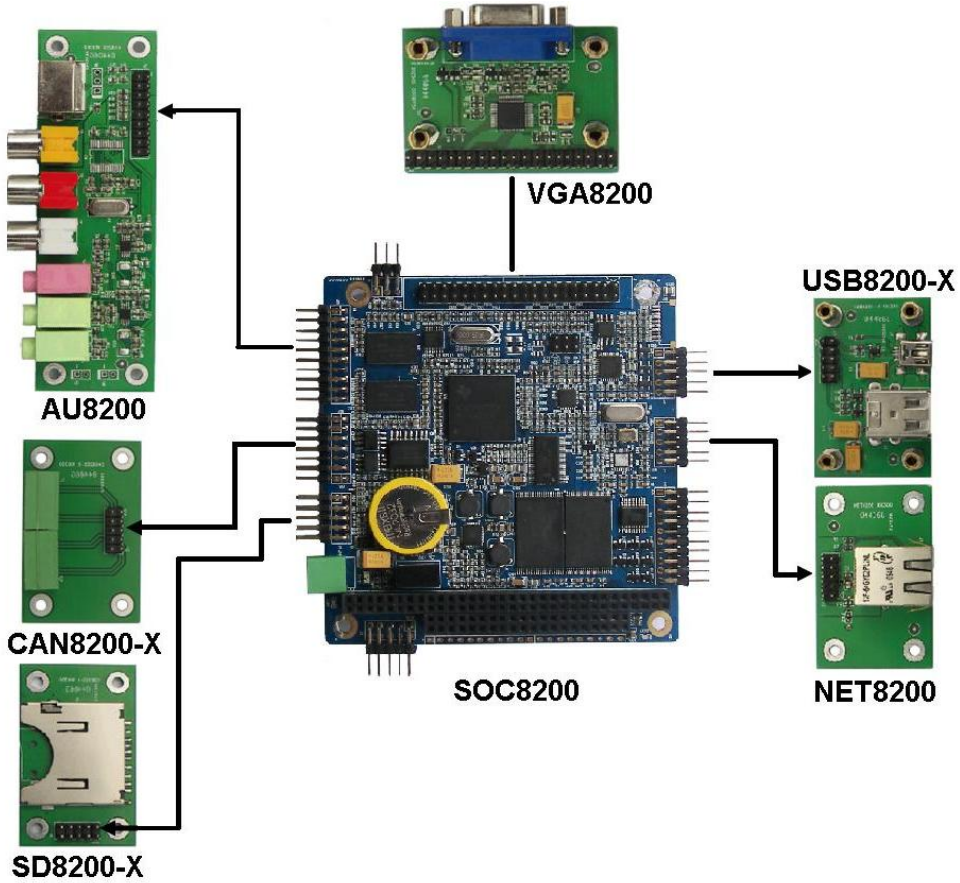
- Audio input interface
- Stereo audio output interface
- 15 PIN standard VGA interface
- Buzzer, output

Transport interface:

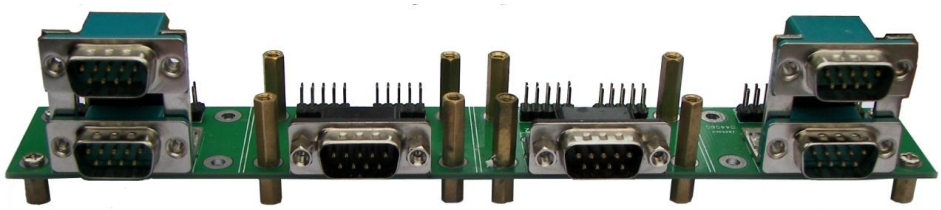
- 1 line, 5 lines serial connectors, RS232 level, DB9 male head
- 1 line, 9 line, RS232 serial port level, DB9 male head
- 1 line, 9 line, TTL level, serial port 2x5 (2.54 mm) row needles interface
- 2 line, USB 2.0 HOST connectors, High-school, 480 Mbps
- 1 line, USB 2.0 Device connectors, High-school, 480 Mbps
- 1 line, 10/100 Ethernet interface, RJ45 connector (with transformer and LED lights)
- 10 bit Camera interface
- Reset button,
- CF card interface
- SD/MMC interface

Appendix 6: Interface Board

For the convenience of our customers and to allow rapid customization of the product, TianMo introduces several interface boards based on the SOC8200 motherboard, each modular function can be connected to the SOC8200 motherboard.



ECOM-4/ECOM-8 (shown below), requires a connection to the E100 communication board for use .



Note:

The interface board is not available through retail, 100PCS minimum.

Appendix 7: ESD Precautions & Handling Procedures

Please note that the board comes without any case/box and all components are exposed. Therefore, extra attention must be paid to ESD (electrostatic discharge) precautions. To effectively prevent electrostatic damage, please follow the steps below:

- Avoid carpets in cool, dry areas. Leave development kits in their anti-static packaging until ready to be installed.

- Dissipate static electricity before handling any system components (development kits) by touching a grounded metal object, such as the system unit unpainted metal chassis.
- If possible, use antistatic devices, such as wrist straps and floor mats.
- Always hold an evaluation board by its edges. Avoid touching the contacts and components on the board.
- Take care when connecting or disconnecting cables. A damaged cable can cause a short in the electrical circuit.
- Prevent damage to the connectors by aligning connector pins before you connect the cable. Misaligned connector pins can cause damage to system components at power-on.
- When disconnecting a cable, always pull on the cable connector or strain-relief loop, not on the cable itself.

Warning:

🕮 This is a class A product. In a domestic environment this product may cause radio interference in which case the user may be required to take adequate measures.

Appendix 8: Technical support & Warranty

Embest Technology Co., Ltd. established in March of 2000, is a global provider of embedded hardware and software. Embest aims to help customers reduce time to market with improved quality by providing the most effective total solutions for the embedded industry. In the rapidly growing market of high end embedded systems, Embest provides comprehensive services to specify, develop and produce products and help customers to implement innovative technology and product features. Progressing from prototyping to the final product within a short time frame and thus shortening the time to market, and to achieve the lowest production costs possible. Embest insists on a simple business model: to offer customers high-performance, low-cost products with the best quality and service.

8.1 Technical support service

Embest provides one year of free technical support for all products. The technical support service covers:

- Embest embedded platform products software/hardware materials
- Assistance to customers with regards to compiling and running the source code we offer.
- Troubleshooting problems occurring on embedded software/hardware platforms if users have followed the instructions provided.
- Judge whether a product failure exists.

The situations listed below are not covered by our free technical support service, and Embest will handle the situation at our discretion:

- Customers encounter issues related to software or hardware during their development process
- Issues occur when users compile/run the embedded OS which has been modified by themselves.

- Customers encounter issues related to their own applications.
- Customers experience problems caused by unauthorised alteration of our software source code

8.2 Maintenance service clause

1. Product warranty will commence on the day of sale and last 12 months provided the product is used under normal conditions
2. The following situations are not covered by the warranty, Embest will charge service fees as appropriate:

- Customers fail to provide valid proof of purchase or the product identification tag is damaged, unreadable, altered or inconsistent with the product.
- Products are subject to damage caused by operations inconsistent with their specification;
- Products are subject to damage in either appearance or function due to natural disasters (flood, fire, earthquake, lightning strike or typhoon) or natural aging of components or other force majeure;
- Products are subject to damage in appearance or function due to power failure, external forces, water, animals or foreign materials;
- Products malfunction due to disassembly or alteration of components by customers, or repair by persons or organizations unauthorized by Embest Technology, or alteration from factory specifications, or configured or expanded with components that are not provided or recognized by Embest Technology;
- Product failures due to the software or systems installed by customers, inappropriate software settings or computer viruses;
- Products purchased from unauthorized merchants;

- Embest Technology takes no responsibility for fulfilling any warranty (verbal or written) that is not made by Embest Technology and not included in the scope of our warranty.

3. Within the period of warranty, the cost for sending products to Embest should be paid by the customer. The cost for returning the product to the customer will be paid by Embest. Any returns in either direction occurring after the warranty period has expired should be paid for by the customer.

4. Please contact technical support with any repair requests.

Note:

📖 Embest Technology will not take any responsibility for products returned without the prior permission of the company.

8.3 Basic guidelines for protection and maintenance of LCDs

1. Do not use finger nails or other hard sharp objects to touch the surface of the LCD

2. Embest recommends purchasing specialist wipes to clean the LCD after long time use, avoid cleaning the surface with fingers or hands as this may leave fingerprints or smudges.

3. Do not clean the surface of the screen with unsuitable chemicals

Note:

📖 Embest do not supply a maintenance service for LCDs. We suggest the customer immediately checks the LCD once in receipt of the goods. In the event that the LCD does not run or shows no display, the customer should inform Embest within 7 business days of delivery.

8.4 Value Added Services

We will provide following value added services:

- Driver development based on Embest embedded platforms for devices such as: serial ports, USB interface devices, and LCD screens.
- Control system transplantation, BSP driver development, API software development.
- Other value added services including supply of power adapters and LCD parts.
- Other OEM/ODM services.
- Technical training.

Please contact Embest with any technical support queries:

http://www.embest-tech.com/contact-us.html